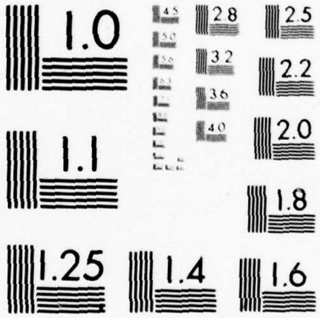
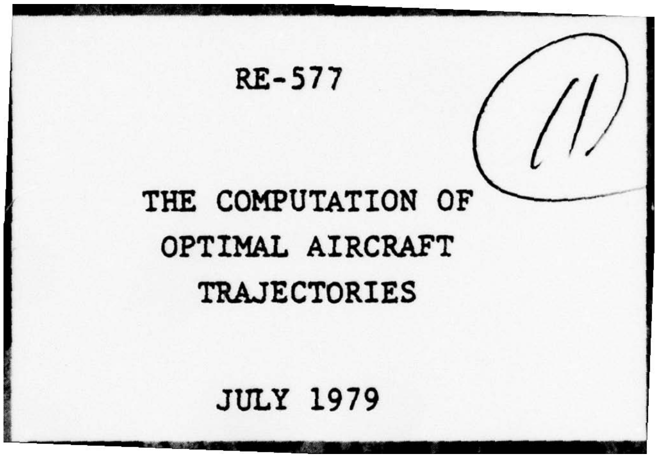
AD-A072 115 GRUMMAN AEROSPACE CORP BETHPAGE N Y RESEARCH DEPT F/G 1/2
THE COMPUTATION OF OPTIMAL AIRCRAFT TRAJECTORIES, (U)
JUL 79 H G MOYER

UNCLASSIFIED RE-577 NL

1 OF 1
AD
A072115

END
DATE
FILMED
9-79
DDC


1.0
4.5
5.0
5.6
6.3
2.8
3.2
3.6
4.0
2.5
2.2
2.0
1.1
1.8
1.25
1.4
1.6



RE-577

11

THE COMPUTATION OF OPTIMAL AIRCRAFT TRAJECTORIES

JULY 1979

U

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER RE-577	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) 6 The Computation of Optimal Aircraft Trajectories.		5. TYPE OF REPORT & PERIOD COVERED 6. PERFORMING ~~ORG. REPORT~~ NUMBER 14 RE-577
7. AUTHOR(s) 10 H.G. Moyer		8. CONTRACT OR GRANT NUMBER(s)
9. PERFORMING ORGANIZATION NAME AND ADDRESS Research Department Grumman Aerospace Corporation Bethpage, New York 11714		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS 11 Jul 79
11. CONTROLLING OFFICE NAME AND ADDRESS 14 RE-577		12. REPORT DATE 7/79 13. NUMBER OF PAGES 55
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office) 12 23p.		15. SECURITY CLASS. (of this report) Unclassified 15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for Public Release; Distribution U nlimited

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Optimal control; aircraft trajectory optimization; conjugate gradient variational tedhnique

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

This report describes an aircraft trajectory optimization computer program that has been used successfully at Grumman for a variety of problems over a period of years. Three airplanes are simulated: the F-14, F-15, and ATF (Advanced Tactical Fighter). The trajectories are in three dimensions over a flat earth. Optimization is achieved by means of the conjugate gradient variational technique. Inequality constraints on the path and equality constraints on the final point are satsified by penalty integrals and penalty functions, respectively.

406 165

UNCLASSIFIED

JOB

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

Grumman Research Department RE-577

THE COMPUTATION OF OPTIMAL AIRCRAFT TRAJECTORIES

by

H. G. Moyer

System Sciences

July 1979

Approved by: Richard A Scheuing
Richard A. Scheuing
Director of Research

ACKNOWLEDGEMENT

The aerodynamic curve-fits and the tabulated data for the engines of the aircraft simulated by the computer programs were prepared by Mr. H. Hinz of the Grumman Research Department. He also provided the curve fits used to model the atmosphere.

ABSTRACT

This report describes an aircraft trajectory optimization computer program that has been used successfully at Grumman for a variety of problems over a period of many years. Three airplanes are simulated: the F-14, F-15, and ATF (Advanced Tactical Fighter). The trajectories are in three dimensions over a flat earth. Optimization is achieved by means of the conjugate gradient variational technique. Inequality constraints on the path and equality constraints on the final point are satisfied by penalty integrals and penalty functions, respectively. The equality constraints can be satisfied to the limit of computer accuracy by a new technique that avoids increases in the penalty function constants.

TABLE OF CONTENTS

NOTATION

a, b, c constants used in the inequality imposed on the path by the structural limit

a_1, b_1 coefficients varying with Mach used in the curve fit for the lift coefficient

a_2, b_2, c_2 coefficients varying with Mach used in the curve fit for the drag coefficient used when $M>0.9$

a_3, b_3 c_3, d_3 coefficients varying with Mach used in the curve fit for the drag coefficient used when $M \leq 0.9$

C_D drag coefficient; see Eqs. (11) and (12)

C_L lift coefficient; see Eq. (10)

C_{L_o} function of Mach appearing in Eq. (12b)

D drag; see Eq. (6) and Fig. 1

f time derivative of the state vector X; see Eq. (21)

g acceleration of gravity

$\hat{g}$ equations for afterburner blow-out and engine operational limit are expressed generically by $\hat{g}\ (M,z) = 0$

$\tilde{g}$ inequalities, Eqs. (7), (8), and (9), imposed on the path and the mixed state-control inequality, Eq. (4), are expressed generically by $\tilde{g}\ (X,u) = 0$

h Heavyside unit step function $h\ (\cdot)$ is zero for negative arguments and one for positive arguments

H generalized Hamiltonian; see Eq. (27)

$\bar{H}$ the true Hamiltonian, which is obtained from the generalized Hamiltonian by setting u to the value that maximizes the latter function

K	diagonal matrix of positive constants; see Eq. (36b)
L	lift; see Eq. (5) and Fig. 1
m	aircraft mass
$\bar{m}$	constant appearing in the limit on the normal aerodynamic force; see Eq. (4)
M	Mach; see Eq. (7)
M_{max}	limit imposed on Mach; see Eq. (7)
n	constant appearing in the limit on the normal aerodynamic force; see Eq. (4)
P	cost functional; see Eq. (23)
P^A	cost functional for a problem with constraints on the final point
P^B	P^A augmented with penalty terms; see Eq. (36)
P^C	cost functional for the fixed final time approximation to a problem with constraints on the final point; see Eq. (42)
S	aircraft reference area
t	time
T	thrust
u	control vector composed of α, μ, and η
v	velocity magnitude
v_s	speed of sound; see Eqs. (16), (18), and (20)
x, y, z	rectangular coordinates of the aircraft, z is the altitude
X	vector of state variables
X_o, X_f	initial and final values of state vector as stated in the problem formulation

α	angle of attack (deg.); see Fig. 1
α'	$\alpha + \epsilon$ (deg)
α_{max}	upper limit on α (deg); see Eq. (3)
β	engine mass flow varying with M, Z, and η
β_u	vector of coefficients used in Eq. (32b); see Eq. (33)
β_τ	coefficient appearing in Eq. (32b); see Eq. (33)
γ	flight path angle measured in the vertical plane between the velocity and the horizontal; see Fig. 1
Δ	vector of positive constants appearing in Eq. (53)
ΔC_L	see Eq. (12b)
ΔD	sum of the ram, interference, and spillage drags in the F-15 engine
ϵ	fixed angle between thrust and fuselage reference line (deg); see Fig. 1
η	throttle coefficient appearing in Eqs. (1d) through (1g)
θ	normalized time; see Eq. (24)
$\bar{\lambda}$	Lagrange multipliers associated with the state vector X
λ_x	adjoint variables associated with the state vector X
λ_τ	adjoint variable associated with the free parameter τ regarded as a state variable
μ	bank angle; the angle between the lift vector L and the vertical plane containing the velocity vector
ρ	atmospheric mass density; see Eqs. (15, (17), and (19)
σ	variable coefficient that scales the increments in u; see (Eq. (32)
τ	final time

τ^A, τ^B, τ^C final times for problems A, B, and C; see Table 1

χ heading angle - the angle between the x axis and the projection of the velocity vector on the horizontal plane

$(\ )^T$ transpose of a vector

$(\dot{\ })$ derivative of the variable with respect to t

$||\ ||_K^2$ generalized norm of a vector; see Eq. (36b)

$\delta(\ )$ variation with the independent variable fixed

$\Delta(\ )$ variation with the independent variable not fixed

$(\ )\epsilon$ ( ) belongs to

$\forall(\ )$ for all ( )

1. INTRODUCTION

This report describes a computer program that calculates optimal three-dimensional aircraft trajectories over a flat earth. The cost functional and terminal constraints can be defined at will. At present there are three FORTRAN decks simulating the F-14, F-15, and ATF (Advanced Tactical Fighter). A fourth program simulates the evasion of a missile by an F-14.

The optimization is accomplished by the conjugate gradient variational algorithm (Ref. 1). Constraints on the final point are satisfied by a new technique that avoids increasing the penalty function constants (Ref. 2). As with other optimization techniques, there is a possibility of converging to a local rather than the absolute minimum. Whenever a gradient-type algorithm is used the results should be checked by computing the problem twice using different starting trajectories. This not only tests for local minimums but also reveals whether or not the "noise" inherent in numerical methods has prevented convergence.

Numerical experiments on the rate of convergence for two-point boundary value problems are described in Ref. 2. Some optimal F-14 maneuvers are presented in Ref. 3.

Descriptions of Grumman programs for paths over a spherical earth may be found in Refs. 4, 5, and 6. The method of Refs. 4 and 5 optimizes trajectories in a vertical plane by means of gradients (rather than conjugate gradients) and satisfies constraints on the final point by increasing the penalty function constants. The method of Ref. 6 optimizes three-dimensional trajectories by converting the variational problem to one with an ordinary minimum.

2. SIMULATION OF THE F-14 TURBOFAN AIRCRAFT

DYNAMICAL EQUALITY AND INEQUALITY CONSTRAINTS

The three-dimensional flight of the F-14 turbofan aircraft is governed by the following differential equations (Ref. 7, p. 48, p. 50). Aircraft and thrust sideslip angles are neglected and rigid body dynamics are not included. The Earth is assumed to be flat. Forces and angles are shown in Fig. 1.

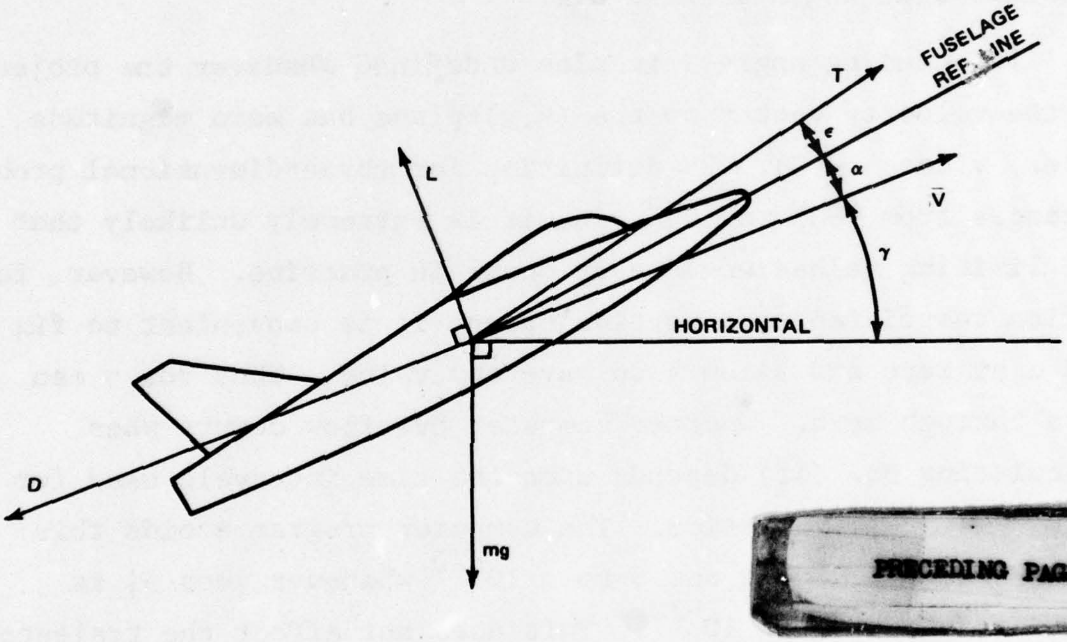


Fig. 1 Forces and Angles in Vertical Plane when Bank Angle Equals Zero

$$\dot{x} = v \cos \gamma \cos \chi \qquad (1a)$$

$$\dot{y} = v \cos \gamma \sin \chi \qquad (1b)$$

$$\dot{z} = v \sin \gamma \qquad (1c)$$

$$\dot{v} = \frac{1}{m} (\eta \; T \cos \alpha' - 1/2 \; \rho \; v^2 S C_D) - g \sin \gamma \tag{1d}$$

$$\dot{\gamma} = \frac{\cos \mu}{m \; v} (\eta \; T \sin \alpha' + 1/2 \; \rho \; v^2 S C_L) - \frac{g}{v} \cos \gamma \tag{1e}$$

$$\dot{\chi} = \frac{\sin \mu}{mv \cos \gamma} (\eta \; T \sin \alpha' + 1/2 \; \rho \; v^2 S C_L) \tag{1f}$$

$$\dot{m} = -\beta \tag{1g}$$

Figure 1 shows that the flight path angle γ is undefined when the magnitude v of the velocity vector is zero. This is associated with the vanishing of the denominator of Eq. (1e). Thus the initial v must not be specified as exactly zero for problems that begin at take-off.

The heading angle χ is also undefined whenever the projection of the velocity vector on the (x,y)-plane has zero magnitude (i.e., $v \cos \gamma = 0$). By definition for three-dimensional problems γ ranges from -90° to $+90^{\circ}$, and it is extremely unlikely that the limiting values would ever occur in practice. However, for motion restricted to a vertical plane it is convenient to fix χ and μ at zero and allow γ to have any value. Thus $\cos \gamma$ can pass through zero. Whether computer overflow occurs when calculating Eq. (1f) depends upon the time intervals used for the numerical integration. The computer program avoids this overflow by resetting $\cos \gamma$ to $\pm 10^{-20}$ whenever $|\cos \gamma|$ is originally less than 10^{-20}. This does not affect the trajectory since, as mentioned above, χ is not obtained from Eq. (1f).

The control variables are the bank angle μ, thrust coefficient η, and angle of attack α (which is an argument of C_L (M,α) and $\alpha' = \alpha + \epsilon$). The latter two control variables must satisfy the inequalities

$$0 \leq \eta \leq 1 \tag{2}$$

$$\alpha \leq \alpha_{max} \tag{3}$$

The limit on the normal aerodynamic force imposes a mixed state-control inequality.

$$L \cos \alpha + D \sin \alpha \le mgn \quad \text{if } m \le \bar{m} \tag{4a}$$

$$L \cos \alpha + D \sin \alpha \le \bar{m}gn \quad \text{if } m > \bar{m} \tag{4b}$$

Here the lift L and drag D are

$$L = 1/2 \; \rho \, v^2 \, SC_L \tag{5}$$

$$D = 1/2 \; \rho \, v^2 \, SC_D \tag{6}$$

and the constant n is usually chosen to be between 6 and 8. Mach number is defined and limited by

$$M = v/v_s \le M_{max} \tag{7}$$

The structural limit imposes a second state inequality, which is modeled as

$$M \le a + bz + cz^2 \tag{8}$$

Finally, the altitude must, of course, be above sea level.

$$z \ge 0 \tag{9}$$

ENGINE CHARACTERISTICS

The thrust T = T(M,z) is obtained either from the augmented thrust table, the military thrust table, or set to zero, depending upon the position of the aircraft relative to the afterburner blow-out altitude and the engine operational limit. The data for both altitudes are curve-fitted as parabolic functions of Mach.

The fuel flow $\beta = \beta(M,z,\eta)$ is nominally trivariate, but at present has been coded only for $\eta = 1$ and only for afterburner thrust. This fuel flow has been used in approximation for trajectories that have short periods either with $\eta<1$ or with military thrust. The data for the engine characteristics were obtained from Refs. 8 and 9. The interpolation procedure is described in Section 9.

AERODYNAMICS

The lift coefficient $C_L(M,\alpha)$ is curve-fitted using

$$C_L = a_1(M) + b_1(M)\ \alpha \tag{10}$$

The coefficients a_1 and b_1 are obtained from tables using linear interpolation.

When $M > 0.9$, the drag coefficient $C_D(M,C_L)$ is modeled as

$$C_D = a_2(M) + b_2(M)C_L + c_2(M)\ C_L{}^2 \tag{11}$$

At lower values of M the relation is

$$C_D = a_3(M) + b_3(M)\ (\Delta C_L)^2 + c_3(M)\ (\Delta C_L)^4 + d_3(M)\ (\Delta C_L)^6 \tag{12a}$$

$$\Delta C_L = C_L - C_{L_o}(M) \tag{12b}$$

Since the Mach-sweep program was operative during the tabulation of the original data (Ref. 10), there is no need for a wing-sweep variable.

3. MODIFICATIONS REQUIRED FOR THE ATF AIRCRAFT

This section presents the changes made in the F-14 program to simulate the ATF (Advanced Tactical Fighter). The fuel flow $\beta(M, z, \eta)$ has been tabulated for both afterburner and military thrust. In each case there are nine values of the throttle setting η. The interpolation is performed by the specially coded subroutine BVI3, which calls on the subroutine BVISP described in Section 9. Interpolation is linear in the z and η directions and by spline in the M direction.

The data for thrust and fuel flow were provided by C. Giannetto of Engineering's Navigation, Guidance, and Control Section. The values of both T and β must be multiplied by three because the ATF is equipped with this number of engines. These data must also be multiplied by the pressure ratio $\rho(z)\, v_s^{\,2}(z)/\rho(o)v_s^{\,2}(0)$. In addition, the fuel flow data are multiplied by $v_s(z)/v_s(0)$. The engine operational limit is 60,000 ft for both military and afterburner power.

The formula for C_L differs from Eq. (10) only in that $a_1(M) = 0$, i.e., $C_L = b_1(M)\alpha$. The equation for C_D has the form of Eq. (12) with $C_{L_o}(M) = d_3(M) = 0$ and with the store drag ΔC_D added.

$$C_D = a_3(M) + b_3(M)C_L^2 + c_3(M)C_L^4 + \Delta C_D(M) \qquad (13)$$

This equation is used for all values of M.

The upper limit α_{max} on the angle of attack is no longer constant, but is tabulated as a function of Mach. As presently coded, the trajectories begin with military thrust and light the afterburner at a later time. The latter is given by $t = \theta_{AB}\tau$

where θ_{AB} is a predetermined constant and τ is the final time. The adjoint variables (see Section 7) do not jump on encountering a discontinuity in the system equations that are defined in this way.

4. MODIFICATIONS REQUIRED FOR THE F-15

Although both the F-14 and the McDonnell-Douglas F-15 use turbofan engines, the ram, interference, and spillage drags cannot be neglected when simulating the latter airplane. The sum of these drags has been tabulated as $\Delta D(M,z)$. Net thrust is obtained by a vectorial subtraction of ΔD from gross thrust. Since ΔD has the same direction as D (Fig. 1), Eq. (1d) is replaced with

$$\dot{v} = (\eta T \cos \alpha' - 1/2 \rho v^2 S C_D - \Delta D)/m - g \sin \gamma \qquad (14)$$

Equations (1e) and (1f) are unchanged in form, although T now represents gross thrust.

The present program has afterburner thrust data only for $\eta = 1$ and has no data at all for military thrust. Thus the engine operational limit coincides with afterburner blow-out. This altitude is expressed either as a linear function of Mach or as 75,000 ft, whichever is lower. The equation for the drag coefficient C_D has the form of Eq. (12) with $d_3(M) = 0$. The same equation is used for all values of M. The upper limit on α is tabulated as a function of Mach.

5. F-14 (EVADER) VERSUS A MISSILE (PURSUER)

The F-14 program has been combined with a missile program (Ref. 11) to simulate missile evasion. The control variables of the F-14 are chosen so that it attempts to evade a pursuing missile. The latter's strictly deterministic trajectory is governed by proportional navigation. The final time associated with the starting trajectories is chosen to be so small that the F-14 cannot be overtaken. At this final time, which is held fixed, the distance separating the vehicles is maximized. The minimum separation as the vehicles move along the final trajectories is found using parabolic interpolation/extrapolation. The final time is then given a small increment toward the time of this minimum, and the process is repeated. The iteration converges to the trajectories that either maximize the minimum separation or maximize the time of impact.

As of the date of this report, the full simulation of the missile trajectory as given in Ref. 11 has not been completed for the above program.

6. ATMOSPHERE

Three regimes are used to approximate the data of Ref. 12 for the mass density ρ (slugs/ft^3) and the speed of sound v_s (ft/sec). The following equations are used when the altitude is below 36,146 ft.

$$\rho = 2.37688 \cdot 10^{-3}(1-6.7911 \cdot 10^{-6}z)^{4.3085} \tag{15}$$

$$v_s = 1116.45(1-6.863956 \cdot 10^{-6}z)^{1/2} \tag{16}$$

For altitudes between 36,146 and 65,874 ft the expressions are

$$\rho = 3.9792633 \cdot 10^{-3} \exp(-4.7829648 \cdot 10^{-5}z) \tag{17}$$

$$v_s = 968.08 \tag{18}$$

The regime for higher altitudes is modeled as

$$\rho = 5.1526166 \cdot 10^{-3}(1+1.6606526 \cdot 10^{-6}z)^{-32.838989} \tag{19}$$

$$v_s = 922.5793652(1+1.535633914 \cdot 10^{-6}z)^{1/2} \tag{20}$$

7. OPTIMIZATION PROCEDURE

The choice of an optimization algorithm was largely determined by the presence of the tabulated functions. This ruled out second order methods (e.g., second variation, quasilinearization) because the numerical approximations to the various second order partial derivatives are unreliable. The classical indirect and the MIN H methods could not be used because it is not possible to express explicitly the value of α that maximizes the generalized Hamiltonian (defined in Eq. (27)). To avoid these difficulties the gradient method was tried at first. Later the extension to conjugate gradient (Ref. 1) was made in an attempt to reduce computer time. A reduction by a factor of about five was obtained.

CONJUGATE GRADIENT ALGORITHM

This subsection presents the conjugate gradient procedure for problems whose final point and final time can be varied freely. On defining the state vector $X^T \equiv (x,y,z,v,\gamma,\chi,m)$ and the control vector $u^T \equiv (\alpha, \mu, \eta)$ the dynamic equations (Eq. (1)) can be written as

$$\dot{X} = f(X,u) \tag{21}$$

This system is to be transferred from the initial point

$$X(o) = X_o \tag{22}$$

to the final time τ and final point $X(\tau)$ that minimize the cost functional

$$P = P(X(\tau),\tau) \tag{23}$$

An important example of this subsection's problem is that of maximizing the altitude ($P = -z$) with the final values of the other components of X as well as the final time open.

In order to keep u(t) as smooth as possible, it is advisable to avoid extrapolation by replacing t with

$$\theta = t/\tau \tag{24}$$

whose range is always 0 to 1. The final time τ could now be regarded as a free parameter; however, the analysis will be a little clearer if it is instead regarded as an additional state variable. If X continues to represent only the original state variables, the system equation takes on the form

$$dX/d\theta = \tau\ f\ (X,u) \tag{25a}$$

$$d\tau/d\theta = 0 \tag{25b}$$

Note that τ is distinguished from the other state variables in that its initial value is not subjected to a constraint of the Eq. (22) type.

The conjugate gradient algorithm employs adjoint variables λ_x, λ_τ that obey

$$\frac{d}{d\theta}\ (\lambda_x^T \delta X + \lambda_\tau \delta\tau) = \tau\ \lambda_x^T\ \frac{\partial f}{\partial u}\ \delta u = \frac{\partial H}{\partial u}\ \delta u \tag{26}$$

The second member has been simplified by defining the generalized Hamiltonian as

$$H \equiv \tau\ \lambda_x^T\ f\ (X,u) \tag{27}$$

The differential equations that λ_x, λ_τ must satisfy when Eq. (26) holds can be found by expanding the left member of this equation and using the equations of variation of Eq. (25).

$$d\lambda_x/d\theta = -(\partial H/\partial X)^T \tag{28a}$$

$$d\lambda_\tau/d\theta = -\partial H/\partial\tau \tag{28b}$$

When the final values of the adjoint variables are defined as

$$\lambda_x(1) = -(\partial P/\partial X)^T \quad , \quad \lambda_\tau(1) = -\partial P/\partial\tau \tag{29}$$

the integral of Eq. (26) becomes

$$\begin{aligned} -\delta P &= \lambda_x^T \delta X + \lambda_\tau \delta\tau \Big|_{\theta=0} + \int_0^1 \frac{\partial H}{\partial u}\,\delta u\,d\theta \\ &= \lambda_\tau(0)\,\delta\tau + \int_0^1 \frac{\partial H}{\partial u}\,\delta u\,d\theta \end{aligned} \tag{30}$$

The last member assumes

$$\delta X(0) = 0 \tag{31}$$

since all the trajectories obey the initial conditions of Eq. (22). Equation (30) indicates that the increments

$$\delta\tau = \sigma\lambda_\tau(0) \quad , \quad \delta u(\theta) = \sigma\left(\frac{\partial H}{\partial u}\,(\theta)\right)^T \tag{32a}$$

are in the direction of steepest descent. The best distance to move in this direction can be found from a search with σ. Although the first descent must be in this direction, the later descents can be in the conjugate gradient direction given by

$$\delta\tau_j = \sigma_j\left[\lambda_{\tau_j}(0) + \beta_\tau\left(\frac{\delta\tau}{\sigma}\right)_{j-1}\right] \ , \ \delta u_j = \sigma_j\left[\left(\frac{\partial H}{\partial u}\right)^T_j + \beta_u\left(\frac{\delta u}{\sigma}\right)_{j-1}\right] \tag{32b}$$

Here the subscript j represents the descent number and β_τ and β_u stand for

$$\beta_\tau = \lambda_{\tau_j}^{2}(0)/\lambda^2_{\tau_{j-1}}(0) \ , \ \beta_u = \left[\int_o^1 \left(\frac{\partial H}{\partial u}\right)^2_{j-1} d\theta\right]^{-1} \int_o^1 \left(\frac{\partial H}{\partial u}\right)^2_j d\theta \tag{33}$$

Although the conjugate gradient theory for problems with n state variables calls for recycling after n descents, experiments with the present variational problem indicate that Eqs. (32a) and (32b) should be alternated from descent to descent for best results.

Whenever Eq. (32) makes $\alpha(\theta) + \delta\alpha(\theta) > \alpha_{max}$ the new α is reset to α_{max} in accordance with Ineq. (3). Similar steps keep η between 0 and 1.

The conjugate gradient algorithm consists of the following steps.

(a) Guess $u(\theta)$ and τ

(b) Compute $X(\theta)$ by integrating Eq. (25). Find P_o from Eq. (23)

(c) Compute $\lambda_x(\theta)$, $\lambda_\tau(\theta)$ by integrating Eq. (28) backward from Eq. (29)

(d) Obtain four values of σ using Eq. (32a) with $\theta = 0$ while $\delta\alpha$, $\delta\mu$, $\delta\eta$, $\delta\tau$ separately take on small values. Set σ_1 equal to the smallest of the σ's

(e) Increment $u(\theta)$ and τ using Eq. (32) with σ_1

(f) Compute $X(\theta)$ by integrating Eq. (25). Find P_1 from Eq. (23)

(g) If $P_1 > P_o$, $\sigma_1/10 \rightarrow \sigma_1$. If the new σ_1 is above a very small tolerance return to step (e), otherwise terminate the computation

(h) Set $\sigma_2 = 4\sigma_1$ and find P_2 as in steps (e) and (f)

(i) Set σ to the value at the minimum of the parabola through $(0,P_o)$, (σ_1,P_1), (σ_2,P_2)

$$\sigma = \left[\frac{\sigma_1}{2}\right]\left[\frac{(4^2-1)P_o-4^2P_1+P_2}{(4-1)P_o-4P_1+P_2}\right]$$

(j) Obtain a set of four values of σ as in step (d) but using large though acceptable values of $\delta\alpha$, $\delta\mu$, $\delta\eta$, and $\delta\tau$. Set σ_3 to the smallest of these and the σ of step (i). Compute P_3 as in steps (e) and (f)

(k) Set P_o to the smallest of P_1, P_2, P_3. Use the corresponding σ in Eq. (32) to update u,τ. If $P_o \neq P_3$, recompute $X(\theta)$ so that the new nominal trajectory is in storage

(l) Return to step (c)

TWO-POINT BOUNDARY VALUE PROBLEMS

This subsection shows that the solution to a problem that specifies constraints on the final point can be approached by a sequence of variable endpoint problems (Ref. 2). First the time-optimal case will be treated.

The original problem -- to be called Problem A (see Table 1) -- is to transfer the system obeying the differential equations, Eq. (25), and the initial conditions, Eq. (22), to

$$X = X_f \tag{34}$$

via the path that minimizes the cost functional

$$P^A = \tau \tag{35}$$

The final time defined by the solution extremal will be called τ^A.

TABLE 1 PROBLEM FORMULATIONS

Problem	Cost Functional	Initial Conditions	Final Conditions
A	τ	$X = X_o$, τ open λ_x open, $\lambda_\tau = 0$	$X = X_f$ λ_x open, $\lambda_\tau = -1$
B	$\tau + \frac{1}{2}\|\|X - X_f\|\|^2_K$	same as above	$\lambda_x = -K(X - X_f)$ $\lambda_\tau = -1$
C	$\frac{1}{2}\|\|X - X_f\|\|^2_K$	$X = X_o$, τ fixed λ_x, λ_τ open	τ open, $\lambda_\tau = 0$ $\lambda_x = -K(X - X_f)$

The algorithm begins with a preliminary stage whose purpose is merely to find an optimal trajectory that is fairly close to the solution and that has a lower final time. This is accomplished by solving Problem B (Table 1), which differs from Problem A in that the final constraints, Eq. (34), are ignored and the cost functional, Eq. (35), is replaced with

$$P^B = \tau + \tfrac{1}{2}||X(1) - X_f||^2_K \tag{36a}$$

The penalty term is defined as

$$||X-X_f||^2_K \equiv (X-X_f)^T K(X-X_f) \tag{36b}$$

where K is a diagonal matrix of positive constants.

After Problem B has been solved by the procedure of the previous subsection, the results can be used to obtain a first order estimate of τ^A. Equation (26) is integrated assuming only $\partial H(\theta)/\partial u = 0$.

$$\lambda_x^T \delta X + \lambda_\tau \delta\tau \Big|_{\theta=0} = \lambda_x^T \delta X + \lambda_\tau \delta\tau \Big|_{\theta=1} \tag{37}$$

This important relation holds between the endpoints of any extremal and those of any adjacent arc whose $\delta X(\theta)$, $\delta\tau$, $\delta u(\theta)$ obey the equations of variation of Eq. (25). Let us evaluate it for the extremal of Problem B using the endpoints of the (otherwise unknown) extremal of Problem A. This implies

$$\delta X(0) = 0 \quad , \quad \delta X(1) = X_f - X^B(1) \tag{38}$$

so that Eq. (37) can be written as

$$\delta\tau = \lambda_x^T(1) \left[X_f - X^B(1)\right] / \left[\lambda_\tau(0) - \lambda_\tau(1)\right] \tag{39}$$

In accordance with Eqs. (29) and (36)

$$\lambda_x^T(1) = -\left[X(1) - X_f\right]^T K \tag{40}$$

so that the first order estimate for the difference between the final times of the two extremals is

$$\delta\tau = \left[X^B(1)-X_f\right]^T K \left[X^B(1)-X_f\right] / \left[\lambda_\tau^B(0)-\lambda_\tau^B(1)\right] \qquad (41)$$

It can be shown with the aid of Eq. (59) that $\delta\tau$ is positive.

The algorithm now proceeds to the second stage which works with Problem C (Table 1). The cost functional

$$P^C = 1/2||X(1) - X_f||^2_K \qquad (42)$$

is to be minimized with τ kept fixed at the following first-order estimate for τ^A.

$$\tau^C = \tau^B + \delta\tau \qquad (43)$$

The solution procedure of the Conjugate Gradient Algorithm subsection is used, except that $\delta\tau = 0$ replaces the first members of Eqs. (32a) and (32b) and the second member of Eq. (29) requires $\lambda_\tau(1) = 0$ (see Table 1).

The above formulation of the variable endpoint Problem C has been chosen so that its solution is identical to that of Problem A, provided τ^C is exactly equal to τ^A. If τ^C happens to be smaller than τ^A, the boundary value errors obtained from the solution to Problem C may exceed the tolerances. A new estimate for τ^A can then be obtained from $\tau^C + \delta\tau \rightarrow \tau^C$ with $\delta\tau$ determined by Eq. (41) using $X^C(1)$, $\lambda_\tau^C(0)$, $\lambda_\tau^C(1)$ instead of $X^B(1)$, $\lambda^B{}_\tau(0)$, $\lambda_\tau^B(1)$.

The τ^C defined by Eq. (43) might also be too large. In this case the algorithm converges to a nonextremal trajectory that makes the boundary value errors negligible. There is then no way of estimating the magnitude of $\tau^C-\tau^A$. Therefore the following algorithm recommends that the increment in τ given by Eq. (41) be divided by two as a safety measure.

The algorithm is summarized in the following steps:

(a) Compute a starting trajectory. Set the descent counter j to zero

(b) Compute a descent cycle for Problem B. $j + 1 \rightarrow j$

(c) Compute $\delta\tau$ from Eq. (41)

(d) If $j < 5$, go to step (b)

(e) If the conjugate gradient direction decreases τ (see Eq. (32)), go to step (b)

(f) If the values of $\tau + \delta\tau$ computed from the three most recent descents do not agree to within a given tolerance, go to step (b)

(g) Compute a new starting trajectory with $\tau + \delta\tau/2 \rightarrow \tau$. $j = 0$

(h) Compute a descent cycle for Problem C. $j + 1 \rightarrow j$

(i) Terminate the computation if all boundary value errors lie within given tolerances

(j) Compute $\delta\tau$ from Eq. (41)

(k) If $j < 5$, go to step (h)

(l) If $\delta\tau < 0$, go to step (h)

(m) If the values of $\delta\tau$ computed from the three most recent descents do not agree to within a given tolerance, go to step (h)

(n) Go to step (g)

The numbers five and three in steps (d), (f), (k), and (m) were chosen arbitrarily. Step (e) was inserted to help the program converge to a τ^B that is below τ^A. Step (l) helps test for convergence because $\delta\tau$ approaches a positive value when

$\tau^C < \tau^A$ and a negligible value when $\tau^C > \tau^A$.

These methods are easily extended to problems other than that of minimum time. For example, suppose the cost functional is $X_n(1)$ with the final values of the other state variables specified. The final time can be either fixed or open. Let us define X_n^A as the value of $X_n(1)$ on the solution extremal. If X_n^A were known, the gradient program could generate the solution extremal by minimizing the error norm, Eq. (32b) with $X_{n_f} = X_n^A$. But a first-order approximation for X_n^A can be found from Eq. (37) using the adjoint variables defined by the extremal that minimizes

$$P = X_n(1)+1/2 \sum_{i=1}^{n-1} K_i \left[X_i(1)-X_{i_f}\right]^2 \qquad (44)$$

Numerical results for F-14 trajectories are presented in Refs. 2 and 3.

DISCONTINUITIES IN THE SYSTEM EQUATIONS

If at a time t the afterburner blows out or the engine reaches its operational limit, $\dot{X}$ jumps from $f^- = f(t^-)$ to $f^+ = f(t^+)$. If the nominal arc were an extremal, the vector equation that governs the jumps in the Lagrange multipliers $\bar{\lambda}$ would be well known (Ref. 13, p. 104).*

$$\bar{\lambda}^+ = \bar{\lambda}^- + \frac{(\bar{\lambda}^{T-}(f^- - f^+))}{(\mathcal{g}_x \ f^+)} \left(\frac{\partial \mathcal{g}}{\partial X}\right)^T \qquad (45)$$

Here $\mathcal{g}(X) = 0$ is the hypersurface at which the system equations, Eq. (21), are discontinuous (see Engine Characteristics in Section

*Literal subscripts indicate partial differentiation in this subsection. The independent variable is once more t rather than θ.

2). Equation (45) is nonlinear in $\bar{\lambda}^+$ since f^+ depends on u^+ which for extremals is a function of $\bar{\lambda}^+$. Taking the scalar product of both sides with f^+ shows that the Hamiltonian $\bar{H} = \bar{\lambda}^T f$ is continuous.

The present problem is slightly different. The proper behavior must be determined for adjoint variables λ as they are integrated along arcs with $H_u \neq 0$ and u previously defined. The theory of the gradient procedure requires the adjoint variables to jump so as to preserve the influence function properties expressed by the following equations (cf. Eq. (30)).

$$P(\tau) = (\partial P(\tau)/\partial X(t'))\ \delta X(t') = -\ \lambda^T(t')\delta X(t') \quad , \ t' \epsilon \left[0,\tau\right] \qquad (46a)$$

$$\delta u(t) = 0, \quad \forall t \in \left[t',\tau\right] \qquad (46b)$$

At the hypersurface of discontinuity

$$\Delta X = \delta X^+ + f^+ \delta t = \delta X^- + f^- \delta t \qquad (47)$$

Here δt is the difference in the times at which the nominal and varied arcs reach the hypersurface $\hat{g} = 0$ and the $\delta X^{\pm}$ are evaluated with time fixed. Thus ΔX lies in the hypersurface tangent space.

$$\hat{g}_x \Delta X = \hat{g}_x \delta X^- + \hat{g}_x f^- \ \delta t = 0 \qquad (48)$$

Eliminating δt between Eqs. (47) and (48) yields

$$\delta X^+ = \delta X^- + \frac{(\hat{g}_x \delta X^-)}{(\hat{g}_x f^-)}\ (f^+ - f^-) \qquad (49)$$

The definition of λ^- must indicate that any δX^- has the same effect on P as the corresponding δX^+.

$$-\delta P = \lambda^{T+}\delta X^+ = \lambda^{T-}\delta X^-$$

$$= \lambda^{T+}\left[\delta X^- + \frac{(\hat{g}_x \delta X^-)}{(\hat{g}_x f^-)}\ (f^+ - f^-)\right] \qquad (50)$$

The last two members will balance for an arbitrary δX^- if

$$\lambda^- = \lambda^+ + \frac{(\lambda^{T+}(f^+ - f^-))}{(\hat{g}_x f^-)}\left(\frac{\partial \hat{g}}{\partial X}\right)^T \qquad (51)$$

Here we have solved for λ^- rather than λ^+ because adjoint variables are integrated backward in time. Note that the right side of Eq. (51) is independent of λ^- and that the form of this equation is equivalent to that of Eq. (45) for Lagrange multipliers.

The jump in λ produces a jump in the δu defined by Eq. (32). Thus the u of the converged trajectory should have a jump identical to that that would be obtained if the indirect method were used instead of the conjugate gradient. Note that the adjoint variables are continuous at the boundaries of the atmospheric regimes because f is then continuous (see Section 6).

INEQUALITIES IMPOSED ON THE PATH

The inequalities, Ineqs. (7), (8), and (9), imposed on the state variables will be satisfied using penalty integrals (Refs. 4 and 5). This method is readily incorporated into a gradient program; however, it provides only an approximation to the optimal path. The mixed state-control inequalitiy, Ineq. (4), will also be treated by this method because it cannot be solved explicitly for the control α. These inequalities can be written generically as

$$\tilde{g}_j(X,u) \le 0 \qquad\qquad j = 1,\ldots,4 \qquad (52)$$

The system equations are augmented with penalty state variables that obey*

$$\frac{dX_i}{d\theta} = \frac{\tau}{2}\,(\tilde{g}_j+\Delta_j)^2 h(\tilde{g}_j+\Delta_j), \quad X_i(o) = 0,\ j = i-8,\ i=9,\ldots,12 \qquad (53)$$

Here h is the Heaviside unit step function, and the positive constants Δ_j have been introduced so that the penalty integrals begin accumulating while the trajectory is still in the admissible region.

A given problem is solved at first with the constraints, Ineq. (52), ignored; that is, with the corresponding penalty function constants in the cost functional, Eq. (36), *set to zero. If the* solution violates the constraint $\tilde{g}_j$, a final value is assigned to $x_i(1)$ $(i = j + 8)$. This value is of course smaller than that computed for the unconstrained trajectory. The modified problem is then solved with the new boundary condition satisfied by the procedure given earlier in this section under Two-Point Boundary Value Problems. If the constraint, Ineq. (52), is still violated, the problem must be solved again either with a smaller final value assigned to the penalty state variable $X_i(1)$, or with the Δ_{i-8} made more positive, or with a larger penalty function constant.

Note that the right sides of Eq. (53) are continuous even though they contain step functions, so that by the theory of the preceeding subsection λ is also continuous when $\tilde{g}_j + \Delta_j = 0$. The analysis (Ref. 14) for exactly optimal paths also requires λ

*The original components of the state vector are the seven variables of Eq. (1) plus τ.

to be continuous for three of the present inequalities, but not at the junctions between interior and boundary arcs of the "second order" constraint $z \geq 0$. But, then, the penalty integral technique also uses a differing differential equation for λ. Since this technique employs optimization theory to bring in the constraints, we may be confident that it yields a good approximation to the exact minimum.

INTERCEPTION OF A MOVING TARGET

This subsection presents the modifications in the solution procedure for the two-point boundary value problem that are required when the final point is in motion; that is, when X_f is a given function of τ rather than a constant (see Problem A in Table 1 earlier in this section). As before, the original problem will be solved by means of Problems B and C whose cost functionals now have the arguments

$$P^B = P^B(X(1), X_f(\tau), \tau), \quad P^C = P^C(X(1), X_f(\tau)) \tag{54}$$

For Problem C, X(o) and τ(o) are both fixed and $u(\theta)$ is to be chosen so that X(1), τ(1) minimize P^C. Of course, we know that $u(\theta)$ has no effect on τ(1), but transversality conditions are assigned as though the computer does not know this. The point is of no significance since we shall see that the relevant quantity in Problem C is $\lambda_\tau(o) - \lambda_\tau(1)$, and this is independent of $\lambda_\tau(1)$. Thus for both Problems B and C

$$\lambda_\tau(1) = -\frac{\partial P}{\partial \tau} - \frac{\partial P}{\partial X_f}\frac{dX_f}{d\tau} \tag{55}$$

The second of Eq. (38) is replaced with

$$\delta X(1) = X_f(\tau) - X(1) + (dX_f/d\tau)\ \delta\tau(1) \tag{56}$$

so that Eq. (39) changes to

$$\delta\tau = \frac{\lambda_x^T(1)\ \left[X_f - X(1)\right]}{\lambda_\tau(o) - \lambda_\tau(1) - \lambda_x^T(1)\ dX_f/d\tau} \tag{57}$$

The derivation of this equation does make use of the relation $\delta\tau(1) = \delta\tau(o)$.

When the nominal trajectory has converged to an extremal with a fixed τ (Problem C), it can be shown using the integral, Eq. (59), of the next subsection that

$$\delta\tau = \left.\frac{\lambda_x^T\ (X_f - X)}{\lambda_x^T\ (f - dX_f/d\tau)}\right|_{\theta=1} \tag{58}$$

This expression is valid only when the numerator is an order of magnitude smaller than the denominator. However, some moving-target problems specify some of the components of f and $dX_f/d\tau$ to be identical. The denominator of Eq. (58) then has a tendency to be small so that Eq. (57) is unreliable. Such problems can be treated by replacing the δτ of Eq. (57) with a fixed, small positive constant. However, this means that the optimal final time is found only approximately and at considerably more computational expense.

FIRST INTEGRALS

The state variable τ is of course constant as are the adjoint variables associated with x, y, and the four penalty state variables introduced above in the subsection <u>Inequalities Imposed on the Path</u>. For trajectories that have converged to extremals, the Hamiltonian $H = \tau \sum_{i=1}^{12} \lambda_i f_i$ is constant. The equation $d\lambda_\tau/d\theta = -H/\tau$ then has the integral

$$\lambda_\tau + H\theta/\tau = \lambda_\tau + (\lambda_x^T f)\theta = \text{const.} \tag{59}$$

Finally the quantity $\lambda_y x - \lambda_x y + \lambda_\chi$ is also constant along extremals (Ref. 15). The last three integrals are not used by the computer program although the time history of H is printed in order to verify convergence.

8. NUMERICAL INTEGRATION

The differential equations are integrated numerically using a modified Adams predictor-corrector method. The procedure is illustrated for the vector set of equations $\dot{X} = f(X)$. The corrector equation is

$$X_{k+1} = X_k + \frac{\Delta t}{2} \left[f(X_k) + f(Y_{k+1})\right] \qquad (60)$$

There are two corrections at each time interval. For the first correction Y_{k+1} is set to the predicted value

$$Y_{k+1} = X_k + \frac{\Delta t}{2} \left[3f(X_k) - f(X_{k-1})\right] \qquad (61)$$

At the initial time $f(X_{-1})$ is assumed to be equal to $f(X_o)$. For the second correction Y_{k+1} is set to the X_{k+1} obtained from the first correction. Variations in the time interval Δt are not permitted by the present program because values from the forward integration for the state variables are required during the backward integration for the adjoint variables.

9. STORAGE AND INTERPOLATION OF TABULATED FUNCTIONS

AFTERBURNER THRUST

The afterburner thrust $T = T(M,z)$ data have been key-punched in the format of the Grumman Data Systems Software Library subroutine TABIN (Index: 12.7.0.2). The subroutine itself is not used to read in the cards because it is coded in single precision. The thrust and its partial derivatives with respect to M and z are obtained from the specially coded subroutine BVISP. Spline interpolation with respect to Mach is used along the curves of constant altitude. The equations are in the form given in Ref. 16. The second derivatives at the mesh points are found at the beginning of the main program and then stored for later use by the interpolation subroutine. If the thrust goes to zero as M decreases to some value called $\bar{M}$, then the second derivatives as well as the values of T are set equal to zero at the mesh points of the region $M \in [0,\bar{M}]$. After T has been found on the two curves that straddle the aircraft's altitude, linear interpolation with respect to z is used to obtain the thrust of the vehicle.

The FORTRAN statement used to obtain afterburner thrust T(M,z) is

```
CALL BVISP (NO, N1, N2, L1, L2, TAB, M,z, SP3, DTDM, DTDZ, T)
```

NO is the fixed point number that has been assigned to each table. For afterburner thurst NO is 1.

N1 is the block of storage containing the number of first arguments of each bivariate table. N1(1) contains the number of Machs tabulated for afterburner thrust.

N2 is the block of storage containing the number of second arguments of each table. N2(1) contains the number of altitudes tabulated for afterburner thrust.

L1 is the storage block that contains the first arguments of each table. For example, L1(1,2) contains the second Mach of the afterburner thrust table.

L2 is the storage block that contains the second arguments of each table. For example, L2(1,2) contains the second altitude of the afterburner thrust table.

TAB is the storage block that contains the values of each table. For example, TAB(3,2,1) contains the afterburner thrust corresponding to the third altitude and the second Mach.

M is the current value of the first argument.

z is the current value of the second argument.

SP3 is the storage block that contains $\partial^2T/\partial M^2$ at each mesh point of the afterburner thrust table. For example, SP3(1,2) contains $\partial^2T/\partial M^2$ at the second mesh point of the curve for the first altitude.

DTDM is the output $\partial T/\partial M$.

DTDZ is the output $\partial T/\partial z$.

T is the output afterburner thrust.

FUEL FLOW AND MILITARY THRUST

The fuel flow $\beta(M,z,\eta)$* and military thrust $T(M,z)$ have

*Recall that fuel flow has been key-punched only for $\eta=1$ except for the ATF. Section 3 presents the interpolation for the latter aircraft.

also been key-punched in the format of subroutine TABIN (see the previous subsection, Afterburner Thrust). Regions for which data are unavailable are indicated by zeros. The interpolation is planar and is computed by the specially coded subroutine BVI. If values are defined at all four enclosing mesh points, the one that is furthest from the nominal point is discarded and the remaining points are used to determine the interpolating plane. In the other cases, the three closest mesh points that have defined values and are not collinear are used.

No significant changes in the trajectories were observed when afterburner thrust was obtained by spline rather than the original planar interpolation. Since the trajectories are even less sensitive to the tables of this subsection, their interpolation was kept planar.

The FORTRAN statement used to obtain either fuel flow or military thrust is

```
CALL BVI (NO, N1, N2, L1, L2, TAB, M, z, DADM, DADZ, A)
```

The arguments are the same as those of BVISP except that SP3 is omitted. NO is 4 for fuel flow and 5 for military thrust.

10. REFERENCES

1. Lasdon, L.S., Mitter,S.K., and Waren, A.D., "The Conjugate Gradient Method for Optimal Control Problems", IEEE Tranactions on Automatic Control, Vol. AC-12, pp. 132-138, 1967.

2. Moyer, H.G., "Improved Gradient Algorithm for Two-Point Boundary Value Problems," AIAA J.,Vol. 13, pp. 17-19, 1975.

3. Hinz, H.K., and Moyer, H.G., "Optimum Zoom Climb Characteristics of the F-14A," Grumman Research Department Memorandum RM-588, 1974 (Confidential).

4. Kelley, H.J., Falco, M., and Ball, D.J., "Air Vehicle Trajectory Optimization," Grumman Research Department Report RE-161J-5, 1962.

5. Kelley, H.J., and Falco, M., "Aircraft Symmetric Flight Optimization," Controls and Dynamic Systems: Advances in Theory and Applications, Vol. 10, pp. 89-129, C. T. Leondes, ed., Academic Press, New York, 1973.

6. Stettler, M., and Voss, J.W., "Formulation and Application of a Trajectory Optimization Program for Powered Lifting Vehicles," Grumman Advanced Development Report No. FSR-AD6-06-68.3, 1968.

7. Miele, A., "Flight Mechanics I - Theory of Flight Paths," Addison-Wesley, Reading, Massachusetts, 1962.

8. "TF30-P-412 Estimated Installed Performance (High Altitude Data)," Propulsion Data Release 303-635, Grumman Aerospace Corporation, November 3, 1972.

9. "F-14A/TF30-P-412A Installed Propulsion System Performance -- Maximum Afterburning and Non-Augmented," Flight Sciences Section Data Release No. FAD-303-1-0-Va. 881, Grumman Aerospace Corporation, May 15, 1973 (Confidential).

10. "Demonstration Data Report, Aerodynamic Demonstration Performance, Section 3," Report No. A51-70-R-71-5, Grumman Aerospace Corporation, July 1, 1973 (Confidential).

11. Hinz, H., and Moyer, G., "Airborne Countermeasures Simulations Study - End Game Analysis," Grumman Research Department Report RE-530, 1976.

12. U. S. Standard Atmosphere - 1962, Superintendent of Documents, December 1962.

13. Bryson, A., and Ho, Y.-C., Applied Optimal Control, Blaisdell, Waltham, Massachusetts, 1969.

14. Jacobson, D.H., Lele, M.M., and Speyer, J.L., "New Necessary Conditions of Optimality for Control Problems with State-Variable Inequality Constraints," J. of Mathematical Analysis and Applications, Vol. 35, pp. 255-284, 1971.

15. Vinh, N.X., "Integrals of the Motion for Optimal Trajectories in Atmospheric Flight," AIAA J., Vol. 11, pp. 700-703, 1973.

16. Mummolo, F., and Lefton, L., "Cubic Splines and Cubic Spline Lattices for Digital Computation," Report No. 72-28, Analytical Mechanics Associates, Jericho, New York, 1972.

APPENDIX

COMPUTER PROGRAM LISTING AND TYPICAL SOLUTION

The FORTRAN IV Computer program listing is reproduced on the following pages with all classified statements and data replaced by asterisks. The particular problem is the minimum time to climb of the F-14 from the end of the runway to z(altitude) = 55,000 feet and V (velocity) = 1548.8 feet/second, with z constrained to be always positive. The numerical solution to this problem follows the program listing.

For those interested in seeing the classified version of the above problem, the program listing, including propulsion and aerodynamic tables, is available from the author.

```
/*          PUNCH LOCAL
C       CLASRIFIED INFORMATION HAS BEEN REPLACED WITH ASTERISKS
C       H.G. MOYER - MAY 1973
C       OPTIMAL TRAJECTORIES FOR THE F14
         IMPLICIT REAL *8(A-H,L,M,O-Z)
        DIMENSION N1(10),N2(10),L1T(5,25),L2T(5,20),TAB(20,25,5)
        DIMENSION FUL(2),ABB(2),AB(25),BB(25),A(20),CON(20),TFB(3)
        DIMENSION CV(5,210),X(7,210),XD(20),XDD(20),XI(20),XC(20)
       1,HMU(210),XB(20),P(10),SB(10),             XF(20),  HMUS(210),
       2 PTB(5),      H(210) ,HAL(210), HALS(210) ,HETA(210),HETAS(210)
        DIMENSION SP1(25),SP2(25),SP3(20,25)
         SIN(X)=DSIN(X)
         COS(X)=DCOS(X)
         ABS(X)=DABS(X)
         SIGN(X,Y)=DSIGN(X,Y)
        DATA XF/20*0.D0/,   A/20*0./,H/210*0./
C       TARLES FOR LIFT COEFFICIENT - CL=AR(M)+BB(M)*ALPHA
         DATA AB/*****
         DATA BB/*****
         TFBC=7.D-3
         TFBC=1.D-3
         TFBC=0.
         TFBC=2.D-2
         TFBC=1.D-2
C       MAX NO. OF INCREMENTS IN THE FINAL TIME
         SHL=1.5D0
         SHL=.5D0
         SHL=3.5D0
        SHL=20.D0
         SHL=8.5D0
C       PENALTY FUNCTION CONSTANTS
         A(3)=1.D-6
         A(4)=1.D-3
         A(4)=2.D-4
         A(10)=0.
         A(10)=5.D-11
         A(11)=1.D-2
         A(11)=0.
C       EMIN=1. PRFCLUDES PARTIAL THRUSTING
        EMIN=0.D0
        EMIN=1.
        EMAX=1.D0
C       NUMBFR OF INTEGRATTON POINTS
        EN=201.
        DT=1.D0/(EN-1.)
        N=FN+.1D0
C       NO OF STATF VARIABLES
          NE=11
C       TRUE CANOPY PLACARD
        MCHLS=*****
C       TRUE LOWER LIMIT ON ALTITUDE
        ZLS=0.D0
        G=32.174D0
        RD=57.29577951D0.
C       THRUST CANT (DEG.)
         ET=*****
C       REFFRENCE AREA
        S=565.D0
        F2=S/2.D0
C       READ THRFE TABLER
```

```
      DO 50 JTN=1,3
      READ (5,22) ITN,N1(ITN),N2(ITN)
      N1O=1
      N1F=MINO(9,N1(ITN))
   52 READ (5,23) (L1T(ITN,I),I=N1O,N1F)
      N2F=N2(ITN)
      DO 51 I=1,N2F
C     TABLES FOR FUEL FLOW AND AFTERBURNER AND MILITARY THRUST
   51 READ (5,24)L2T(ITN,I),(TAB(I,J,ITN),J=N1O,N1F)
      N1O=N1O+9
      IF(N1O.GT.N1(ITN)) GO TO 50
      N1F=MINO(N1F+9,N1(ITN))
      GO TO 52
   50 CONTINUE
        DO 69 I=1,25
        DO 69 J=1,12
   69   TAB(J,I,4)=TAB(J,I,4)*2.
      READ (5,22) ITN,N1(3),N2(3)
      N1O=2
      N1F=MINO(8,N1(3))
   62 READ (5,25) (L1T(3,I),I=N1O,N1F)
      DO 61 I=1,8
C     TABLE FOR CD
   61 READ (5,25) (TAB(I,J,3),J=N1O,N1F)
      N1O=N1O+7
      IF(N1O.GT.N1(3)) GO TO 60
      N1F=MINO(N1F+7,N1(3))
      GO TO 62
   60 CONTINUE
      L1T(3,1)=0.
      DO 63 I=1,8
   63 TAB(I,1,3)=TAB(I,2,3)
 1200  CONTINUE
C     INITIAL VALUES
      READ(5,10) (XI(I),I=1,7)
        XI(5)=XI(5)/RD
        XI(6)=XI(6)/RD
C     FINAL VALUES
      READ(5,10) (XF(I),I=1,7),TF
C     ANGLE OF ATTACK AND BANK ANGLE
C     PROGRAM WORKS WITH ALPHA IN DEGREES, MU IN RADIANS
      READ(5,20) (CV(1,I),I=1,N)
        DO 400 I=1,N
        IF(I.GE.185) CV(1,I)=CV(1,185)
        CV(2,I)=0.
        CV(5,I)=1.
  400  CONTINUE
C     PRELIMINARY WORK FOR SPLINE INTERPOLATION TO BE USED LATER
C     FOR AFT. THRUST
      J=1
      DO 1250 K=1,20
      DO 1260 I=1,23
      SP3(K,I)=0.
      IF(TAB(K,I,J).NE.0.D0) GO TO 1261
 1260 CONTINUE
 1261 IO=I+1
      SP2(IO)=4.D0
      DO 1251 I=IO,24
      SP1(I)=6.D0*(TAB(K,I-1,J)-2.*TAB(K,I,J)+TAB(K,I+1,J))
      IF(I.EQ.IO)GO TO 1251
      SP1(I)=SP1(I)-SP1(I-1)/SP2(I-1)
 1252 SP2(I)=4.-1./SP2(I-1)
 1251 CONTINUE
C     FIND THE SECOND DERIVATIVES USED FOR SPLINE INTERPOLATION
      SP3(K,25)=0.
      I=24
 1253 SP3(K,I)=(SP1(I)-SP3(K,I+1))/SP2(I)
      I=I-1
      IF(I.GE.IO)GO TO 1253
 1250 CONTINUE
C     WORKING CANOPY PLACARD
       MCHL=*****
C     WORKING LOWER LIMIT ON ALTITUDE
       ZL= 50.D0
C     SET COUNTERS FOR DECENT AND FINAL TIME SHIFTS
      SH=0.
       SH=1.
      D=0.D0
C     BEGIN PRINTING AT TIME ZERO
       NP=1
  310  DO 290 I=1,N
C     INITIALIZE FOR CONJUGATE GRADIENT STEP
      HALS(I)=0.D0
      HETAR(I)=0.
  290  HMUS(I)=0.D0
      HALTS=1..D20
       HMUTS=1.D20
      HETATS=1.D20
       LTFOS=1.D20
  241 INDS=7
C     INITIALIZE INTEGRATION CYCLE
      J=1
       IF(D.EQ.0.D0) WRITE (6,14)
      DO 125 I=1,N
       IF(CV(1,I).GT.9.4)  CV(1,I)=9.4D0
      CV(3,I)=SIN(CV(2,I))
  125 CV(4,I)=COS(CV(2,I))
      DO 101 I=1,7
      X(I,1)=XI(I)
  101 XC(I)=XI(I)
       DO300 I=8,20
  300  XC(I)=0.
C     INITIALIZE MAXIMUM LOAD FACTOR AND STRUCTURAL LIMIT
      SL=-1.D20
      LW=-1.D20
      HAR=-1.D20
      HEL=-1.D20
  112 M=XC(7)
      Z=XC(3)
      V=XC(4)
      SG=SIN(XC(5))
      CG=COS(XC(5))
       IF(CG.GE.0.)  CG=DMAX1(CG,1.D-20)
       IF(CG.LE.0.)  CG=DMIN1(CG,-1.D-20)
      SCHI=SIN(XC(6))
      CCHI=COS(XC(6))
      AL=CV(1,J)
      SMU=CV(3,J)
      CMU=CV(4,J)
          SE=SIN((AL+ET)/RD)
      CE=COS((AL+ET)/RD)
C     OBTAIN AIR DENSITY
      IF(Z.LE.36146.D0) RHO=2.37688D-3*(1.-.67911D-5*Z)**4.3085D0
       IF(Z.GE.36146.D0.AND.Z.LE.65874.D0)
     1 RHO= 3.0792633D-3*DEXP(-.4782968D-4*Z)
      IF(Z.GT.65874.D0) RHO=5.1526166D-3/(1.+.16606526D-5*Z)** 32.838989
     1D0
C     OBTAIN VELOCITY OF SOUND
       VS=968.08
      IF(Z.LT.36146.) VS=1116.45D0*DSQRT(1.-6.863956D-6*Z)
      IF(Z.GT.65874.) VS=   922.5793652D0*DSQRT(1.+1.5356339140D-6*Z)
      F1=F2*V/M*RHO
      MCH=V/VS
      THR=0.
      IF(Z.GT.*****                                  GO TO 860
C     INTERPOLATE LINEARLY WITH RESPECT TO ALTITUDE, BY SPLINE
C     WITH RESPECT TO MACH TO GET AFT. THRUST
      CALL BVISP(1,N1,N2,L1T,L2T,TAB,MCH,Z,SP3,DTDM,DTDZ,THR)
C     SHOULD MILITARY THRUST BE USED?
      IF(Z.GT.*****
C     PLANAR INTERPOLATION
     1CALL BVI(5,N1,N2,L1T,L2T,TAB,MCH,Z,DTDM,DTDZ,THR)
      THR=THR*CV (5,J)
 860   IF( THR.LT.0.)  WRITE (6,36) MCH,Z
       IF(THR.LT.0.) THR=0.
       I=MCH/.1D0+1.
       I=MINO(I,24)
       C=AR(I)+(AB(I+1)-AB(I))*DMOD(MCH,.1D0)/.1
       B=BR(I)+(RB(I+1)-RB(I))*DMOD(MCH,.1D0)/.1
C     LIFT COEFFICIENT
       CL=C+B*AL
      DO 1000 I=2,24
      IF(MCH-L1T(3,I)) 1001,1001,1000
 1000 CONTINUE
      I=24
 1001 IF( CL.GT..9D0) GO TO 1300
      CM=C+B*AL-TAB(1,I-1,3)
      CD1=TAB(2,I-1,3)+TAB(3,I-1,3)*CM**2+TAB(4,I-1,3)*CM**4
     1+TAB(5,I-1,3)*CM**6
      CN=C+B*AL-TAB(1,I   ,3)
      CD2=TAB(2,I,3)+TAB(3,I,3)*CN**2+TAB(4,I,3)*CN**4+TAB(5,I,3)*CN**6
      GO TO 1301
 1300 CD1=TAB(6,I-1,3)+TAB(7,I-1,3)*CL+TAB(8,I-1,3)*CL*CL
      CD2=TAB(6,I,3)+TAB(7,I,3)*CL+TAB(8,I,3)*CL*CL
 1301 CDMA=(CD2-CD1)/(L1T(3,I)-L1T(3,I-1))
C     DRAG COEFFICIENT
      CD=CD1+CDMA*(MCH-L1T(3,I-1))
C     FUEL FLOW
       BE    =0.
       IF(THR.GT.0.)
     1CALL BVI(4,N1,N2,L1T,L2T,TAB,MCH,Z,BETAM,BETAZ,BE)
       BETA=BE    /3600.D0/G
C     DERIVATIVES OF THE STATE VARIABLES
      XD(1)=V*CG*CCHI
      XD(2)=V*CG*SCHI
      XD(3)=V*SG
       XD(4)=  THR*CE/M-F1*CD*V-G*SG
      XD(5)=F1*CMU*CL-G/V*CG  +THR/M*SE/V *CMU
       XD(6)=  (F1*CL+THR/M*SE/V)*SMU/CG
      XD(7)=-BETA
```

```
        XD(A)=0.
C      CANOPY.PLACARD.
       IF(MCH.GT.MCHL) XD(8)=(MCH-MCHL)**2/2.D0
       MS=*****.
        XD(9)=0.
C      STRUCTURAL LIMIT
       IF(MCH.GT.MS) XD(9)=(MCH-MS)**2/2.
        XD(10)=0.
C      ALTITUDE.LIMIT
        IF(Z.LT.ZL)   XD(10)=(Z-ZL)**2/2.
C       LOAD FACTOR
          LFL=*****
        IF(G*M.GT.*****       LFL=*****
        XD(11)=0.
           IF(F1*CL*V/G.GT.LFL)   XD(11)=(F1*CL*V/G-LFL)**2/2.
       DO 107 I=1,NE
C      SCALE DERIVATIVES BY THE FINAL TIME
  107  XD(I)=XD(I)*TF
       IF(IND)109,102,103
  102  IND=7
       DO 104 I=1,NE
C      CORRECT
  104  XC(I)=XB(I)+DT*(XD(I)+XDO(I))/2.
       GO TO 112
  103  DO 108 I=1,NE
C      CORRECT AGAIN AND STORE
  108  XC(I)=XB(I)+DT*(XD(I)+XDO(I))/2.
        DO 2108 I=1,7
 2108  X(I,J)=XC(I)
C      TIME
  109  T=(J-1)*TF/(N-1)
       IF(D.NE.0.0) GO TO 1209
        IF(J.EQ.1)  WRITE (6,16)
C      ENERGY
        BE=XC(3)+XC(4)**2/2./G
       LFL=*****
       IF(G*M.GT.*****       LFL=*****
        LW9=F1*V*CL/G-LFL
        LW=DMAX1(LW,LW9)
       ENL=(F1*V*CL+THR/M*SE)/G
       IF(MOD(J, 8).EQ.1)WRITE (6,13) T,THR,Z,MCH,CL,AL,CD,BE,LW9,
     1  ENL,H(J)
 1209  CONTINUE
       SL=DMAX1(SL,*****
       HAB=DMAX1(HAB,*****
       HEL=DMAX1(HEL,*****
       IF(J.EQ.N) GO TO 247
        DO 106 I=1,NE
       XB(I)=XC(I)
        IF(IND.LT.0) XDO(I)=XD(I)
C      PREDICT
        XC(I)=XC(I)+(3.*XD(I)-XDO(I))*DT/2.
  106   XDO(I)=XD(I)
        IND=0
       J=J+1
       GO TO 112
  247  CONTINUE
       IF(D.EQ.0.D0) WRITE (6,14)
       DO 250I=NP,N,8
       T=(I-1)*TF/(N-1)
```

```
      ALD=CV(1,I)
      MUD=RD*CV(2,I)
      GD=RD*X(5,I)
       IF(GD.GT.180.D0)  GO TO 251
C     AVOID DIVIDING BY COS(GAMMA)WHEN GAMMA IS CLOSE TO 90 DEG.
      CHID=RD*X(6,I)
C     PRINT TRAJECTORY
  250 WRITE (6,15) I,ALD,MUD,(X(J,I),J=1,4),GD,CHID,X(7,I) ,CV(5,I)
      IF(D.NE.0.D0) GO TO 1270
       CHIT=XF(6)*RD
      WRITE (6,12)
       WRITE (6,15) (XF(I),I=1,5),CHIT
      QF=RHO*V*V/2.
      WRITE (6,33) SL
      WRITE (6,32) LW
      WRITE (6,34) HAB
      WRITE (6,35) HEL
      WRITE (6,31) QF
      WRITE (6,14)
 1270 CONTINUE
       WRITE (6,13) TFB,(XC(I),I=8,11)
       IF(DMOD(D,91.D0).EQ.9.D0)   GO TO 1201
      IF(SH-SHL)122,1201,1201
 1201  CONTINUE
C     PUNCH
C     WRITE(7,11) (XI(I),I=1,7)
C     WRITE(7,11) (XF(I),I=1,7),TF
C     WRITE(7,21) (CV(1,I),I=1,N)
       GO TO 1200
C     ORIGINAL COST FUNCTIONAL
  122 P(1)=TF
      DO 1322 I=1,NE
      CON(I)=XC(I)-XF(I)
C     INITIAL CONDITIONS FOR LAMBDA INTEGRATION
      XC(I)=-A(I)*CON(I)
C     ADD PENALTIES ON THE FINAL CONDITIONS
 1322 P(1)=P(1)+A(I)/2.*CON(I)**2
 1430  L1=XC(1)
      L2=XC(2)
       L8 =XC( 8)
       L9 =XC(9 )
       L10=XC(10)
       L11=XC(11)
C     INITIALIZE ADJOINT VARIABLE FOR THE FINAL TIME STATE VARIABLE
      XC(8)=-1.
C     INITIALIZE INTEGRATION CYCLE
      J=N
      IND=-7.
      HALT=0.
      HMUT=0.
      HETAT=0.
  142 SG=SIN(X(5,J))
      CG=COS(X(5,J))
       IF(CG.GE.0.)  CG=DMAX1(CG,1.D-20)
       IF(CG.LE.0.)  CG=DMIN1(CG,-1.D-20)
      SCHI=SIN(X(6,J))
      CCHI=COS(X(6,J))
      M=X(7,J)
      Z=X(3,J)
      V=X(4,J)
      SMU=CV(3,J)
      CMU=CV(4,J)
       AL=CV(1,J)
         SE=SIN((AL+ET)/RD)
         CE=COS((AL+ET)/RD)
      ETA=CV(5,J)
      V2=V*V
      IF(Z.LE.36146.D0) RHO=2.37688D-3*(1.-.67911D-5*Z)**4.3085D0
      IF(Z.LE.36146.D0) DRDZ=.67911D-5*4.3085*RHO/(.67911D-5*Z-1.)
       IF(Z.GE.36146.D0.AND.Z.LE.65874.D0)
    1 RHO= 3.9792633D-3*DEXP(-.47829648D-4*Z)
       IF(Z.GE.36146.D0.AND.Z.LE.65874.D0) DRDZ=-.47829648D-4*RHO
      IF(Z.GT.65874.D0) RHO=5.1526166D-3/(1.+.16606526D-5*Z)** 32.838989
      IF(Z.GE.65874.D0) DRDZ=-.16606526D-5*32.838989D0*RHO
     1/(1.+.16606526D-5*Z)
       VS=968.08
       DVSDZ=0.
      IF(Z.LT.36146.) VS=1116.45D0*DSQRT(1.-6.863956D-6*Z)
      IF(Z.LT.36146.) DVSDZ=-1116.45**2   *6.863956D-6/2./VS
      IF(Z.GT.65874.) VS=   922.5793652D0*DSQRT(1.+1.535633914D-6*Z)
      IF(Z.GT.65874.) DVSDZ=922.5793652D0**2    *1.535633914D-6/2./VS
      F1=F2*V/M*RHO
      MCH=V/VS
      J2=MOD(J,2)+1
      ABR(J2)=*****
      EOL(J2)=*****
      THR=0.D0
      TMT=0.
      TZT=0.
      IF(EOL(J2).GE.0.) GO TO 900
      CALL BVISP(I,N1,N2,L1T,L2T,TAB,MCH,Z,SP3,TMT,TZT   ,TH )
      IF(Z.GT.*****
     1CALL BVI(S,N1,N2,L1T,L2T,TAB,MCH,Z,TMT,TZT   ,TH )
      TMT=TMT*ETA
      TZT=TZT*ETA
      THR=TH*ETA
 900   IF(THR.LT.0.) THR=0.
       I=MCH/.1D0+1.
       I=MIN0(I,24)
       C=AB(I)+(AB(I+1)-AB(I))*DMOD(MCH,.1D0)/.1
       B=BB(I)+(BB(I+1)-BB(I))*DMOD(MCH,.1D0)/.1
       CL=C+B*AL
       CLAL=B
       CLMA=(AB(I+1)-AB(I)+(BB(I+1)-BB(I))*AL)/.1
      DO 1002 I=2,24
      IF(MCH-L1T(3,I)) 1003,1003,1002
1002 CONTINUE
      I=24
1003 IF( CL.GT..9D0) GO TO 1302
      CM=C+B*AL-TAB(1,I-1,3)
      CD1=TAB(2,I-1,3)+TAB(3,I-1,3)*CM**2+TAB(4,I-1,3)*CM**4
     1+TAB(5,I-1,3)*CM**6
      CDCL1=2.D0*TAB(3,I-1,3)*CM+4.D0*TAB(4,I-1,3)*CM**3
     1+6.D0*TAB(5,I-1,3)*CM**5
      CN=C+B*AL-TAB(1,I   ,3)
      CD2=TAB(2,I,3)+TAB(3,I,3)*CN**2+TAB(4,I,3)*CN**4+TAB(5,I,3)*CN**6
      CDCL2=2.*TAB(3,I,3)*CN+4.*TAB(4,I,3)*CN**3+6.*TAB(5,I,3)*CN**5
      GO TO 1304
1302 CD1=TAB(6,I-1,3)+TAB(7,I-1,3)*CL+TAB(8,I-1,3)*CL*CL
      CDCL1=TAB(7,I-1,3)+2.*TAB(8,I-1,3)*CL
      CD2=TAB(6,I,3)+TAB(7,I,3)*CL+TAB(8,I,3)*CL*CL
      CDCL2=TAB(7,I,3)+2.*TAB(8,I,3)*CL
 1304 CDMA=(CD2-CD1)/(L1T(3,I)-L1T(3,I-1))
      CD=CD1+CDMA*(MCH-L1T(3,I-1))
      CDCL=CDCL1+(CDCL2-CDCL1)*(MCH-L1T(3,I-1))/(L1T(3,I)-L1T(3,I-1))
       BE    =0.
       BETAM=0.
       BETAZ=0.
       IF(THR.GT.0.)
     1CALL BVI(4,N1,N2,L1T,L2T,TAB,MCH,Z,BETAM,BETAZ,BE)
       BETA  =BE    /3600./G
       BETAM =BETAM/3600./G
       BETAZ =BETAZ/3600./G
  151 L3=XC(3)
      L4=XC(4)
      L5=XC(5)
      L6=XC(6)
      L7=XC(7)
      LTF=XC(8)
      L12=L1*CCHI+L2*SCHI
      L56=L5*CMU+L6*SMU/CG
C     DERIVATIVES OF THE ADJOINT VARIABLES
      XD(3)=(L56*CL-L4*CD*V)*F1/RHO*DRDZ
     1   -F1*V/VS/VS*CLMA     *DVSDZ*L56+L4*F1*MCH*MCH*DVSDZ
     2*(CDMA     +CDCL     *CLMA      )-L7*(BETAZ-MCH/VS*BETAM*DVSDZ)
     3   +(TZT    -TMT     *DVSDZ/VS*MCH)*(L4*CE+L56/V*SE)/M
      XD(4)=L12*CG+L3*SG+L4*(TMT      /M*CE/VS-F1*MCH*(CDMA     +CDCL     *
     1 CLMA      +2.*CD/MCH))+L5/V2*G*CG+L56*F1/V*(MCH*CLMA     +CL)
     2+L56/M*SE/V*(TMT     /VS-THR/V)-L7/VS*BETAM
      XD(5)=-L12*SG*V+L3*V*CG-L4*G*CG+L5*G/V*SG
     1+ L6*SMU*SG/CG/CG*(F1*CL+THR/M*SE/V)
      XD(6)=(L2*CCHI-L1*SCHI)*V*CG
      XD(7)=F1/M*(L4*CD*V-L56*CL)-THR/M/M*(L4*CE+L56*SE/V)
      XD(8)=L12*V*CG+L3*V*SG+L4*(       -F1*CD*V-G*SG+THR/M*CE)
     1+L56*F1*CL-L5*G*CG/V-L7*BETA +L56*THR/M*SE/V
C     COMPUTE THE INEQUALITY CONSTRAINTS FOR THE STATE VARIABLES
      FM=MCH-MCHL
      IF(MCH.GT.MCHL) XD(3)=XD(3)-L8*FM/VS*MCH*DVSDZ
      IF(MCH.GT.MCHL) XD(4)=XD(4)+L8*FM/VS
       IF(MCH.GT.MCHL)   XD(8)=XD(8)+L8/2.*FM*FM
      MS=******
      FMS=MCH-MS
      IF(MCH.GT.MS) XD(3)=XD(3)-L9*FMS*(MCH/VS*DVSDZ+2.78857D-5
     1+2.*1.17551D-10*Z))
      IF(MCH.GT.MS) XD(4)=XD(4)+L9*FMS/VS
      IF(MCH.GT.MS) XD(8)=XD(8)+L9*FMS*FMS/2.
      IF(Z.LT.ZL) XD(3)=XD(3)+L10*(Z-ZL)
       IF(Z.LT.ZL)   XD(8)=XD(8)+L10/2.*(Z-ZL)**2
        LFL=******
       IF(G*M.GT.*****       LFL=******
       EM=F1*CL*V/G
       IF(EM.LE.LFL)  GO TO 890
       FLF=EM-LFL
       XD(3)= XD(3)+L11*FLF*F2*V2/G/M*(CL*DRDZ-RHO*CLMA*MCH/VS*DVSDZ)
       XD(4)=XD(4)+L11*FLF*EM/V/CL*(2.*CL+MCH*CLMA)
       XD(7)=XD(7)-L11*FLF*EM/M
       XD(8)=XD(8)+L11/2.*FLF*FLF
  890  CONTINUE
      DO 143 I=3,7
  143 XD(I)=XD(I)*TF
```

```
      IF(IND)150,144,145
  144 IND=7
      DO 146 I=3,8
C     CORRECT
  146 XC(I)=XB(I)+DT*(XD(I)+XDO(I))/2.
      GO TO 151
  145 DO 152 I=3,8
C     CORRECT AGAIN AND STORE
  152 XC(I)=XB(I)+DT*(XD(I)+XDO(I))/2.
C     IS THE ENGINE OPERATIONAL LIMIT BEING PASSED?
      IF(EOL(1)*EOL(2).GE.0.) GO TO 902
      THRO=0.
      DV=(*****
      DZ=-DV*MCH*DVSDZ-1.
      GO TO 903
C     IS THE AFTERBURNER BLOWING OUT?
  902 IF(ABB(1)*ABB(2).GE.0.) GO TO 150
      CALL BVI(5,N1,N2,L1T,L2T,TAB,MCH,Z,TMTO,TZTO,THRO)
      DV=*****
      DZ=-DV*MCH*DVSDZ-1.
  903  EK=(CE*
     1,     L4+L56*SE/V)*(THR-THRO)/M/(DZ*V*SG+DV*(CE*THR /M-F1*V*CD-G*SG)
     2)
      IND=-1
C     LAMBDAS 3 AND 4 JUMP WHEN PASSING THROUGH AFTERBURNER BLOWOUT
C     AND ENGINE OPERATIONAL LIMITS
      XC(3)=L3-EK*DZ
      XC(4)=L4-EK*DV
      GO TO 151
C     PARTIAL H PARTIAL MU
  150  HMU(J)=(XC(6)*CMU/CG-XC(5)*SMU)*(F1*CL+THR/M*SE/V)
       L56=  XC(5)*CMU+XC(6)*SMU/CG
       L4=XC(4)
C     PARTIAL H PARTIAL ALPHA
      HAL(J)=F1*CLAL    *(L56-L4*V*CDCL    )
     1+ (L56*CE/V-L4*SE)*THR/M/RD
       IF(EM.GT.LFL)  HAL(J)=HAL(J)+L11*FLF*F1*V/G*CLAL
      HETA(J)=TH/M*(L4*CF+L56*SE/V)
      HALT=HALT+HAL(J)**2
      HMUT=HMUT+HMU(J)**2
      HETAT=HETAT+HETA(J)**2
C     HAMILTONIAN/TF (SHOULD CONVERGE TO A CONSTANT)
       H(J)=XD(8)
      IF(J.EQ.1) GO TO 160
      DO 147 I=3,8
      XB(I)=XC(I)
       IF(IND.LT.0) XDO(I)=XD(I)
C     PREDICT
       XC(I)=XC(I)+(3.*XD(I)-XDO(I))*DT/2.
  147  XDO(I)=XD(I)
      IND=0
      J=J-1
      GO TO 142
  160 CONTINUE
      LTFO=XC(8)
       ID=DMOD(D,3.D0)+1.1D0
       TFB(ID)=TF
       FTF=2.
       FTF=3.
       DO 1420 I=1,NF
```

```
C     STORE THF 3 LATEST ERTIMATES FOR THE CORRECT FINAL TIME
 1420  TFB(ID)=A(I)*CON(I)**2/(1.+LTFO     )/FTF+TFB(ID)
C     SHOULD THE FINAL TIME BE INCREASED IN ORDER TO BRING
C     IN THE FINAL CONDITIONS?
       IF(D.GT.95.D0)GO TO 243
       IF(D.LE.4.D0.OR.LTFO.LE.-1.D0)  GO TO 1371
       IF(D.GT.25.D0) GO TO 1400
       IF(SH.EQ.0..AND.LTFO.LT.0.)  GO TO 1371
 1400  CONTINUE
       IE(ABS(TFB(1)-TFB(2))+ABS(TFB(2)-TFB(3)) +ABS(TFB(3)-TFB(1))
     1 .LT.TF*TFBC )  GO TO 243
 1371  CONTINUE
       CGF=1.
      HALTS=DMAX1(HALTS,HALT*CGF )
       HMUTS=DMAX1(HMUTS,CGF*HMUT) +1.D-20
      HETATS=DMAX1(HETATS,CGF*HETAT)+1.D-20
       BT=LTFO**2/DMAX1(LTFOS**2,CGF*LTFO**2)
       SIG1=2.D-2*TF/ABS(LTFO+BT*LTFOS)
       I=30
      SIG2=.01D0/ABS(HAL(I)+HALT/HALTS*HALS(I))
       I=1
       SIG3=.1D0/ABS(HAL(I)+HALT/HALTS*HALS(I))
       IF(A(2)+A(6).GT.0.)  SIG3=.1D0/ABS(HMU(1)+HMUT/HMUTS*HMUS(1))
C      SIG IS THE SIZF OF THE STEP IN THE CONJUGATE GRADIENT DIRECTION
       SIG=DMIN1(     SIG1,SIG2,SIG3)
      K=2
  235 DO 161 I=1,N
       MU=CV(2,I)+SIG*(HMU(I)+HMUT/HMUTS*HMUS(I))
       CV(3,I)=SIN(MU)
  161  CV(4,I)=COS(MU)
       TFN=TF
       IF(SH.EQ.0.)
     1 TFN=TF+SIG*(LTFO+BT*LTFOS)
      DO 201 I=1,7
C      INITTIALIZE INTEGRATION CYCLE
  201 XC(I)=XI(I)
       DO301 I=8,20
  301  XC(I)=0.
      IND=-7
      J=1
       IF(TFN) 99,99,212
  212 V=XC(4)
        IF(V.LT.0.)  SIG=SIG/5.
        IF(V.LT.0.)  GO TU 235
      Z=XC(3)
       IF(Z.LT.-1.D4)  SIG=SIG/5.D0
       IF(Z.LT.-1.D4)  GO TO 235
      SG=SIN(XC(5))
      CG=COS(XC(5))
       IF(CG.GE.0.)  CG=DMAX1(CG,1.D-20)
       IF(CG.LE.0.)  CG=DMIN1(CG,-1.D-20)
      SCHI=SIN(XC(6))
      CCHI=COS(XC(6))
      M=XC(7)
      AL=CV(1,J)+SIG*(HAL(J)+HALT/HALTS*HALS(J))
      IF(AL.GT.9.5D0) AL=9.5
      SMU=CV(3,J)
      CMU=CV(4,J)
      ETA=CV(5,J)+SIG*(HETA(J)+HETAT/HETATS*HETAS(J))
      TF(ETA.GT.FMAX) FTA=EMAX
      IF(ETA.LT.EMIN) ETA=EMIN
          SE=SIN((AL+ET)/RD)
          CE=COS((AL+ET)/RD)
C     OBTAIN AIR DENSITY
      IF(Z.LE.36146.D0) RHO=2.376880-3*(1.-.67911D-5*Z)**4.3085D0
       IF(Z.GE.36146.D0.AND.Z.LE.65874.D0)
     1 RHO= 3.9792633D-3*DEXP(-.47829648D-4*Z)
      IF(Z.GT.65874.D0) RHO=5.1526166D-3/(1.+.16606526D-5*Z)** 32.838989
C     OBTAIN VELOCITY OF SOUND
       VS=968.08
      IF(Z.LT.36146.) VS=1116.45D0*DSQRT(1.-6.863956D-6*Z)
      IF(Z.GT.65874.) VS=   922.5793652D0*DSQRT(1.+1.535633914D-6*Z)
      F1=F2*V/M*RHO
      MCH=V/VS
      THR=0.
      IF(Z.GT.*****                                    GO TO 861
C     THRUST
      CALL BVISP(1,N1,N2,L1T,L2T,TAB,MCH,Z,SP3,DTDM,DTDZ,THR)
      IF(Z.GT.*****
     1CALL BVI(5,N1,N2,L1T,L2T,TAB,MCH,Z,DTDM,DTDZ,THR)
       IF(THR.LT.0.) THR=0.
      THR=THR*ETA
  861 CONTINUE
       I=MCH/.1D0+1.
       I=MIN0(I,24)
       C=AB(I)+(AB(I+1)-AB(I))*DMOD(MCH,.1D0)/.1
       B=BB(I)+(BB(I+1)-BB(I))*DMOD(MCH,.1D0)/.1
C     LIFT COEFFICIENT
       CL=C+B*AL
      DO 1004 I=2,24
       IF(MCH-L1T(3,I)) 1005,1005,1004
 1004 CONTINUE
       I=24
 1005 IF(CL.GT..9) GO TO 1303
       CM=C+B*AL-TAB(1,I-1,3)
       CD1=TAB(2,I-1,3)+TAB(3,I-1,3)*CM**2+TAB(4,I-1,3)*CM**4
     1+TAB(5,I-1,3)*CM**6
       CN=C+B*AL-TAB(1,I   ,3)
       CD2=TAB(2,I,3)+TAB(3,I,3)*CN**2+TAB(4,I,3)*CN**4+TAB(5,I,3)*CN**6
      GO TO 1305
 1303 CD1=TAB(6,I-1,3)+TAB(7,I-1,3)*CL+TAB(8,I-1,3)*CL*CL
      CD2=TAB(6,I,3)+TAB(7,I,3)*CL+TAB(8,I,3)*CL*CL
 1305 CDMA=(CD2-CD1)/(L1T(3,I)-L1T(3,I-1))
C     DRAG COEFFICIENT
      CD=CD1+CDMA*(MCH-L1T(3,I-1))
C     FUEL FLOW
       BETA =0.
       IF(THR.GT.0.)
     1CALL BVI(4,N1,N2,L1T,L2T,TAB,MCH,Z,BETAM,BETAZ,BETA)
       BETA=BETA/3600.D0/G
C     DERIVATIVES OF THE STATE VARIABLES
      XD(1)=V*CG*CCHI
      XD(2)=V*CG*SCHI
      XD(3)=V*SG
       XD(4)= THR*CE/M-F1*CD*V-G*SG
      XD(5)=F1*CMU*CL-G/V*CG +THR/M*SE/V *CMU
       XD(6)= (F1*CL+THR/M*SE/V)*SMU/CG
      XD(7)=-BETA
       XD(8)=0.
C     CANOPY PLACARD
```

```
      IF(MCH.GT.MCHL) XD(8)=(MCH-MCHL)**2/2.D0
      MS=******.
       XD(9)=0.
C     STRUCTURAL LIMIT
      IF(MCH.GT.MS) XD(9)=(MCH-MS)**2/2.
       XD(10)=0.
C     LOWER LIMIT ON ALTITUDE
       IF(Z.LT.ZL)   XD(10)=(Z-ZL)**2/2.
        LFL=8.D0-1.
       IF(G*M.GT.*****       LFL=******
       XD(11)=0.
C     LOAD FACTOR LIMIT
         IF(F1*CL*V/G.GT.LFL)  XD(11)=(F1*CL*V/G-LFL)**2/2.
      DO 207 I=1,NE
C     SCALE DERIVATIVES BY THE FINAL TIME
  207 XD(I)=XD(I)*TFN
      IF(IND) 209,202,203
  202 IND=7
      DO 204 I=1,NE
C     CORRECT
  204 XC(I)=XB(I)+DT*(XD(I)+XDO(I))/2.
      GO TO 212
  203 DO 208 I=1,NE
C     CORRECT AGAIN AND STORE
  208 XC(I)=XB(I)+DT*(XD(I)+XDO(I))/2.
       DO 2208 I=1,7
 2208 X(I,J)=XC(I)
       QF=RHO*X(4,N)**2/2.
      IF(J.EQ.N) GO TO 222
  209  DO 206 I=1,NE
      XB(I)=XC(I)
       IF(IND.LT.0) XDO(I)=XD(I)
C     PREDICT
       XC(I)=XC(I)+(3.*XD(I)-XDO(I))*DT/2.
  206  XDO(I)=XD(I)
       IND=0
      J=J+1
      GO TO 212
C     COMPUTE PAYOFF PLUS PENALTY FUNCTIONS
  222 P(K)=TFN
      DO 1325 I=1,NE
 1325 P(K)=P(K)+A(I)/2.*(XC(I) -XF(I))**2
       PTB(K)=P(K)-TFN
      SB(K)=SIG
      IF(K-3) 230,231,232
  230 IF(P(1)-P(2)) 233,233,234
  233 SIG=SIG/10.D1
C     TERMINATE THE COMPUTATION WHEN SIG IS NEGLIGIBLE
      IF(SIG-.1D-9*TF/ABS(LTFO)) 243,243,235
  234 K=3
      SIG=4.*SIG
      GO TO 235
  231 K=4
      FK=SB(3)/SB(2)
C     ESTIMATE OF BEST STEP SIZE FROM PARABOLIC INTERPOLATION
      SIG=SB(2)/2.*(P(3)-P(1)+FK*FK*(P(1)-P(2)))
     1/(P(3)-P(1)+FK*(P(1)-P(2)))
C     DOES THE PARABOLA HAVE A MAX INSTEAD OF A MIN
      IF(SIG.LE.0.) SIG=SB(3)*5.
       SIG1=1.D-1*TF/ABS(LTFO+BT*LTFOS)
       SIG2=.5D0/ABS(HAL(1)+HALT/HALTS*HALS(1))
C     KEEP THE FINAL TIME INCREMENT SMALL
       SIG=DMIN1(SIG,SIG1,SIG2)
      GO TO 235
  232 CONTINUE
      P(1)=DMIN1(P(2),P(3),P(4))
      D=D+1
C     FIND THE BEST STEP IN THE GRADIENT DIRECTION
       K=2
      IF(P(1).EQ.P(3)) K=3
      IF(P(1).EQ.P(4)) K=4
       PT=PTR(K)
       SIG=SB(K)
       IF(SH.EQ.0.)
     1 TF  =TF+SIG*(LTFO+RT*LTFOS)
      DO 240 I=1,N
      AL=CV(1,I)+SIG*(H   (I)+HALT/HALTS*HALS(I))
      IF(AL.GT.9.5D0) A  :9.5
      CV(1,I)=AL
      HALS(I)=HAL(I)
      ETA=CV(5,I)+SIG*(HETA(I)+HETAT/HETATS*HETAS(I))
      IF(ETA.GT.EMAX) ETA=EMAX
      IF(ETA.LT.EMIN) ETA=EMIN
      CV(5,I)=ETA
      HETAS(I)=HETA(I)
       MU=CV(2,I)+SIG*(HMU(I)+HMUT/HMUTS*HMUS(I))
       CV(2,I)=MU
  240  HMUS(I)=HMU(I)
      HALTS=HALT
       HMUTS=HMUT
      HETATS=HETAT
       LTFOS=LTFO
       NP=N
       IF(SH.EQ.0.)   GO TO 1370
C     ARE THE FINAL CONDITIONS SATISFIED?
       IF(PT.LT.A(3)*9.D4)    GO TO 99
 1370  CONTINUE
      SIG=SIG/3.
C     RECOMPUTE THE BEST TRAJECTORY UNLESS IT WAS THE LAST
       IF(K.EQ.4)   GO TO 247
       GO TO 241
  243 D=0.
       TF=DMIN1(TFB(1),TFB(2),TFB(3),TF+1.D1)
       TFN=TF
      SH=SH+1.
       NP=1
      GO TO 310
   99 SH=SHL+1.
       NP=1
       D=0.
       GOTO 241
  251  STOP
   10 FORMAT (8F10.0)
   11  FORMAT (3F10.0,F10.2,2F10.6,F10.2,F10.4)
   12 FORMAT (1H )
   13 FORMAT (1P10D12.4,D11.3)
   14 FORMAT     (/1H )
   15  FORMAT (36X,1P7D12.4)
   16  FORMAT   ( 1H1)
   20  FORMAT (  4D20.13)
```

```
21   FORMAT (1P4D20.13)
22  FORMAT (8X,I4,I2,I2)
23  FORMAT (7X,9D7.0)
24  FORMAT (10D7.0)
25  FORMAT (10X,7D10.0)
31  FORMAT (' Q FINAL =',F9.2,1P5D12.4)
32  FORMAT (' L/W MINUS LOAD FACTOR LIMIT =',F4.1)
33  FORMAT (' MACH MINUS STRUCTURAL LIMIT =',F6.2)
34  FORMAT (' DISTANCE ABOVE AFTERBURNER BLOWOUT HEIGHT =',F8.0)
35  FORMAT (' DISTANCE ABOVE ENGINE OPERATIONAL LIMIT =',F8.0)
36   FORMAT (' NEGATIVE THRUST',1P4D12.4)
    END
```

```
      SUBROUTINE BVI(ITN,N1,N2,L1T,L2T,TAB,A1,A2,TAA,TAC,TA)
C     BI-VARIATE INTERPOLATION AND EXTRAPOLATION
C     PLANAR INTERPOLATION FOR TABLES IN STINT FORMAT
C     INTEVALS MAY BE NONUNIFORM
C     ZEROS SHOULD BE USE FOR UNDEFINED INITIAL OR FINAL SEGMENTS
C     OF INPUT CURVES
C     N1(I)=NO. OF X ARGUMENTS IN TABLE I
C     N2(I)=NO. OF Y ARGUMENTS IN TABLE I
C     L1T(I,J)=X ARGUMENTS FOR TABLE I
C     L2T(I,J)=Y ARGUMENTS FOR TABLE I
C     TAB(J,K,I)=Z VALUES FOR TABLE I
C     A1=X ARGUMENT, A2=Y ARGUMENT
      IMPLICIT REAL *8(A-H,L,M,O-Z)
      DIMENSION N1(10),N2(10),L1T(5,25),L2T(5,20),TAB(20,25,5)
      IIF=N2(ITN)
      DO 611 II=2,IIF
      IF(L2T(ITN,II)-A2) 611,612,612
  611 CONTINUE
      II=IIF
  612 JJF=N1(ITN)
      DO 613 JJ=2,JJF
      IF(TAB(II,JJ,ITN).LE.0..AND.TAB(II,JJ-1,ITN).GT.0.) GO TO 614
      IF(TAB(II,JJ,ITN)) 613,613,609
  609 IF(L1T(ITN,JJ)-A1) 613,614,614
  613 CONTINUE
      JJ=JJF
  614 DO 608 JP=2,JJF
      IF(TAB(II-1,JP,ITN).LE.0..AND.TAB(II-1,JP-1,ITN).GT.0.) GO TO 606
      IF(TAB(II-1,JP,ITN)) 608,608,607
  607 IF(L1T(ITN,JP)-A1) 608,606,606
  608 CONTINUE
      JP=JJF
  606 IF(TAB(II,JJ-1,ITN).GT.0..OR.TAB(II-1,JP-1,ITN).GT.0.) GO TO 605
      IF(JJ-JP) 400,401,402
  400 JJ=JJ+1
      GO TO 604
  401 JJ=JJ+1
  402 JP=JP+1
  605 IF(TAB(II,JJ,ITN).GT.0..OR.TAB(II-1,JP,ITN).GT.0.) GO TO 604
      IF(JJ-JP) 410,411,412
  410 JP=JP-1
      GO TO 604
  411 JP=JP-1
  412 JJ=JJ-1
  604  A11=L1T(ITN,JP-1)
       A12=L1T(ITN,JP)
       A13=L1T(ITN,JJ-1)
       A14=L1T(ITN,JJ)
       A21=L2T(ITN,II-1)
      A22=A21
       A23=L2T(ITN,II)
      A24=A23
       TA1=TAB(II-1,JP-1,ITN)
       TA2=TAB(II-1,JP,ITN)
       TA3=TAB(II,JJ-1,ITN)
       TA4=TAB(II,JJ,ITN)
       IF(TA1) 520,520,510
  510 IF(TA2) 521,521,511
  511 IF(TA3) 522,522,512
  512 IF(TA4) 503,503,513
  513 A1C=(A11+A12)/2.D0
      A2C=(A21+A23)/2.
      IF(A1.GT.A1C.AND.A2.GT.A2C) GO TO 520
      IF(A1.LT.A1C.AND.A2.GT.A2C) GO TO 521
      IF(A1.GT.A1C.AND.A2.LT.A2C) GO TO 522
      GO TO 503
  520 A11=A14
      A21=A24
      TA1=TA4
      GO TO 503
  521 A12=A14
      A22=A24
      TA2=TA4
      GO TO 503
  522 A13=A14
      A23=A24
      TA3=TA4
C     PARTIAL DERIVATIVE WITH RESPECT TO ARGUMENT ONE
  503 TAA=-((A23-A21)*(TA2-TA1)-(A22-A21)*(TA3-TA1))
     1/((A13-A11)*(A22-A21)-(A12-A11)*(A23-A21))
C     PARTIAL DERIVATIVE WITH RESPECT TO ARGUMENT TWO
      TAC=-((TA3-TA1)*(A12-A11)-(TA2-TA1)*(A13-A11))
     1/((A13-A11)*(A22-A21)-(A12-A11)*(A23-A21))
C     INTERPOLATED VALUE
      TA=TA1+TAA*(A1-A11)+TAC*(A2-A21)
      RETURN
      END
      SUBROUTINE BVISP(ITN,N1,N2,L1T,L2T,TAB,A1,A2,SP3,TAA,TAC,TA)
C     INTERPOLATION WITH RESPECT TO ARG1 IS BY SPLINE
C     INTERPOLATION WITH RESPECT TO ARG2 IS LINEAR
C     SP3 CONTAINS THE SECOND DERIVATIVES
      IMPLICIT REAL*8 (A-H,L,M,O-Z)
      DIMENSION N1(10),N2(10),L1T(5,25),L2T(5,20),TAB(20,25,5),SP3(20,25
     1)
      IIF=N2(ITN)
      DO 611 II=2,IIF
      IF(L2T(ITN,II)-A2) 611,612,612
  611 CONTINUE
      II=IIF
  612 JJF=N1(ITN)
      DO 613 JJ=2,JJF
      IF(L1T(ITN,JJ)-A1) 613,614,614
  613 CONTINUE
      JJ=JJF
  614 H=.1D0
      DX1=A1-L1T(ITN,JJ-1)
      DX2=H-DX1
      SP2=SP3(II,JJ)
      SP1=SP3(II,JJ-1)
      TA2=SP1*DX2**3/6.D0/H+SP2*DX1**3/6./H+(TAB(II,JJ-1,ITN)-SP1*H*H/6.
     1)*DX2/H+(TAB(II,JJ,ITN)-SP2*H*H/6.)*DX1/H
      DTDX2=-SP1*DX2**2/2.D0/H+SP2*DX1**2/2./H+(TAB(II,JJ,ITN)
     1-TAB(II,JJ-1,ITN))/H+(SP1-SP2)*H/6.
      SP2=SP3(II-1,JJ)
      SP1=SP3(II-1,JJ-1)
      TA1=(SP1*DX2**3/6.+SP2*DX1**3/6.+(TAB(II-1,JJ-1,ITN)-SP1*H*H/6.)
     1*DX2+(TAB(II-1,JJ,ITN)-SP2*H*H/6.)*DX1)/H
      DTDX1=(SP2*DX1**2/2.-SP1*DX2**2/2.+TAB(II-1,JJ,ITN)
     1-TAB(II-1,JJ-1,ITN))/H+(SP1-SP2)*H/6.
      TAC=(TA2-TA1)/(L2T(ITN,II)-L2T(ITN,II-1))
      TA=TA1+TAC*(A2-L2T(ITN,II-1))
      TAA=DTDX1+(DTDX2-DTDX1)/(L2T(ITN,II)-L2T(ITN,II-1))
     1*(A2-L2T(ITN,II-1))
      RETURN
```

```
    t            T           Z            M            C_L          α°           C_D     ENERGY/mg                                                  H
0.0          4.2220D+04  0.0          3.9998D-01   4.3652D-01   2.9274D+00   3.4324D-02   3.0990D+03  -5.5776D+00   1.0357D+00   3.604D-04
8.5104D+00   4.6252D+04  4.3579D+01   5.6086D-01   2.0028D-01   6.2634D-01   2.6629D-02   6.1349D+03  -5.7067D+00   9.1085D-01   1.252D-02
1.7021D+01   5.1795D+04  6.0896D+01   7.2976D-01   1.8982D-01   6.2438D-01   2.6387D-02   1.0372D+04  -5.1849D+00   1.4695D+00   7.040D-03
2.5531D+01   5.5483D+04  1.2225D+03   8.6597D-01   1.8980D-01   1.4082D+00   2.8208D-02   1.5627D+04  -4.6974D+00   2.0849D+00  -2.492D-02
3.4041D+01   5.3146D+04  4.6562D+03   9.3288D-01   1.3079D-01   9.2340D-01   3.2645D-02   2.1002D+04  -5.3198D+00   1.4173D+00  -3.258D-03
4.2552D+01   4.8254D+04  9.3265D+03   9.5559D-01   6.9288D-02   5.5449D-02   3.3984D-02   2.5882D+04  -6.1086D+00   6.5278D-01   3.210D-02
5.1062D+01   4.3002D+04  1.3631D+04   9.7307D-01   3.2916D-02  -5.0940D-01   3.6028D-02   3.0256D+04  -6.5286D+00   2.6907D-01   4.897D-02
5.9573D+01   4.0157D+04  1.6658D+04   1.0067D+00   2.1550D-02  -7.7521D-01   4.0521D-02   3.4046D+04  -6.6844D+00   1.5389D-01   5.398D-02
6.8083D+01   4.0410D+04  1.7994D+04   1.0690D+00   4.5611D-02  -5.2458D-01   4.4933D-02   3.7395D+04  -6.4758D+00   3.7618D-01   5.240D-02
7.6593D+01   4.3101D+04  1.7996D+04   1.1554D+00   9.1180D-02   1.3456D-01   4.5241D-02   4.0662D+04  -5.9802D+00   9.1165D-01   5.501D-02
8.5104D+01   4.6059D+04  1.7767D+04   1.2414D+00   1.1687D-01   6.0297D-01   4.5520D-02   4.3978D+04  -5.5542D+00   1.3742D+00   5.999D-02
9.3614D+01   4.7429D+04  1.8352D+04   1.3069D+00   1.1479D-01   6.0970D-01   4.5164D-02   4.7270D+04  -5.4972D+00   1.4737D+00   5.995D-02
1.0212D+02   4.7040D+04  2.0022D+04   1.3540D+00   9.8226D-02   3.5854D-01   4.4279D-02   5.0654D+04  -5.7396D+00   1.2657D+00   5.466D-02
1.1063D+02   4.5474D+04  2.2378D+04   1.3929D+00   8.2513D-02   1.0255D-01   4.3511D-02   5.4190D+04  -6.0197D+00   1.0195D+00   4.922D-02
1.1915D+02   4.4197D+04  2.4879D+04   1.4332D+00   6.9523D-02  -3.8621D-02   4.3410D-02   5.7872D+04  -6.2564D+00   8.1721D-01   4.564D-02
1.2766D+02   4.3835D+04  2.7068D+04   1.4812D+00   6.0560D-02  -5.6785D-02   4.3664D-02   6.1669D+04  -6.4163D+00   6.9251D-01   4.635D-02
1.3617D+02   4.3627D+04  2.8644D+04   1.5402D+00   5.6507D-02   6.4959D-02   4.3816D-02   6.5563D+04  -6.4897D+00   6.5556D-01   4.744D-02
1.4468D+02   4.4603D+04  2.9497D+04   1.6102D+00   6.0391D-02   3.8116D-01   4.4090D-02   6.9551D+04  -6.4591D+00   7.4587D-01   4.903D-02
1.5319D+02   4.5736D+04  2.9834D+04   1.6820D+00   7.7027D-02   8.4793D-01   4.5084D-02   7.3411D+04  -6.1927D+00   1.0364D+00   4.424D-02
1.6170D+02   4.6619D+04  3.0283D+04   1.7431D+00   9.7550D-02   1.4206D+00   4.6816D-02   7.6904D+04  -5.8802D+00   1.3971D+00   4.665D-02
1.7021D+02   4.6064D+04  3.1678D+04   1.7843D+00   1.2232D-01   2.0908D+00   4.9314D-02   7.9939D+04  -5.5884D+00   1.7592D+00   4.101D-02
1.7872D+02   4.2966D+04  3.4758D+04   1.8020D+00   1.4400D-01   2.6579D+00   5.1887D-02   8.2650D+04  -5.5483D+00   1.8245D+00   3.549D-02
1.8723D+02   3.4540D+04  3.9613D+04   1.7691D+00   1.4805D-01   2.6861D+00   5.1986D-02   8.5196D+04  -5.9693D+00   1.4467D+00   3.473D-02
1.9574D+02   2.5032D+04  4.5357D+04   1.7024D+00   1.2016D-01   1.8904D+00   4.8405D-02   8.7567D+04  -6.5906D+00   8.2225D-01   3.706D-02
2.0425D+02   1.4216D+04  5.0797D+04   1.6390D+00   1.2326D-01   1.8401D+00   4.8195D-02   8.9892D+04  -6.8510D+00   5.4725D-01   3.736D-02
2.1276D+02   1.3841D+04  5.5323D+04   1.5641D+00   1.2990D-01   1.7844D+00   4.8180D-02   9.0952D+04  -6.9720D+00   4.0721D-01   3.745D-02
    t            α°          μ°           x            y            z            v            γ°           χ°           m            η
0.0          2.9274D+00  0.0          0.0          0.0          0.0          4.4656D+02   0.0          0.0          1.8027D+03   1.0000D+00
8.5104D+00   6.2634D-01  0.0          8.5595D+03   0.0          4.3579D+01   6.2607D+02   3.2798D-01   0.0          1.7935D+03   1.0000D+00
1.7021D+01   6.2438D-01  0.0          1.0690D+04   0.0          6.0896D+01   8.1457D+02   1.8585D+00   0.0          1.7839D+03   1.0000D+00
2.5531D+01   1.4082D+00  0.0          1.8193D+04   0.0          1.2225D+03   9.6275D+02   1.6658D+01   0.0          1.7732D+03   1.0000D+00
3.4041D+01   9.2340D-01  0.0          2.5952D+04   0.0          4.6562D+03   1.0246D+03   2.9929D+01   0.0          1.7629D+03   1.0000D+00
4.2552D+01   5.5449D-02  0.0          3.3390D+04   0.0          9.3265D+03   1.0322D+03   3.2108D+01   0.0          1.7532D+03   1.0000D+00
5.1062D+01  -5.0940D-01  0.0          4.1085D+04   0.0          1.3631D+04   1.0343D+03   2.5479D+01   0.0          1.7445D+03   1.0000D+00
5.9573D+01  -7.7521D-01  0.0          4.9376D+04   0.0          1.6658D+04   1.0576D+03   1.4117D+01   0.0          1.7363D+03   1.0000D+00
6.8083D+01  -5.2458D-01  0.0          5.8494D+04   0.0          1.7994D+04   1.1173D+03   3.2103D+00   0.0          1.7286D+03   1.0000D+00
7.6593D+01   1.3456D-01  0.0          6.8371D+04   0.0          1.7996D+04   1.2077D+03  -1.8600D+00   0.0          1.7205D+03   1.0000D+00
8.5104D+01   6.0297D-01  0.0          7.9445D+04   0.0          1.7767D+04   1.2987D+03   3.0591D-01   0.0          1.7110D+03   1.0000D+00
9.3614D+01   6.0970D-01  0.0          9.0377D+04   0.0          1.8352D+04   1.3641D+03   5.7196D+00   0.0          1.7028D+03   1.0000D+00
1.0212D+02   3.5854D-01  0.0          1.0205D+05   0.0          2.0022D+04   1.4039D+03   1.0156D+01   0.0          1.6955D+03   1.0000D+00
1.1063D+02   1.0255D-01  0.0          1.1388D+05   0.0          2.2378D+04   1.4307D+03   1.1913D+01   0.0          1.6845D+03   1.0000D+00
1.1915D+02  -3.8621D-02  0.0          1.2591D+05   0.0          2.4879D+04   1.4571D+03   1.1204D+01   0.0          1.6757D+03   1.0000D+00
1.2766D+02  -5.6785D-02  0.0          1.3826D+05   0.0          2.7068D+04   1.4922D+03   8.6941D+00   0.0          1.6674D+03   1.0000D+00
1.3617D+02   6.4959D-02  0.0          1.5105D+05   0.0          2.8644D+04   1.5413D+03   5.3062D+00   0.0          1.6593D+03   1.0000D+00
1.4468D+02   3.8116D-01  0.0          1.6441D+05   0.0          2.9497D+04   1.6054D+03   2.1751D+00   0.0          1.6511D+03   1.0000D+00
1.5319D+02   8.4793D-01  0.0          1.7836D+05   0.0          2.9834D+04   1.6746D+03   1.0624D+00   0.0          1.6426D+03   1.0000D+00
1.6170D+02   1.4206D+00  0.0          1.9286D+05   0.0          3.0283D+04   1.7321D+03   3.0121D+00   0.0          1.6337D+03   1.0000D+00
1.7021D+02   2.0908D+00  0.0          2.0769D+05   0.0          3.1678D+04   1.7622D+03   8.2342D+00   0.0          1.6248D+03   1.0000D+00
1.7872D+02   2.6579D+00  0.0          2.2235D+05   0.0          3.4758D+04   1.7555D+03   1.5675D+01   0.0          1.6165D+03   1.0000D+00
1.8723D+02   2.6861D+00  0.0          2.3630D+05   0.0          3.9613D+04   1.7126D+03   2.2233D+01   0.0          1.6090D+03   1.0000D+00
1.9574D+02   1.8904D+00  0.0          2.4940D+05   0.0          4.5357D+04   1.6481D+03   2.4193D+01   0.0          1.6036D+03   1.0000D+00
2.0425D+02   1.8401D+00  0.0          2.6201D+05   0.0          5.0797D+04   1.5780D+03   2.2089D+01   0.0          1.5996D+03   1.0000D+00
2.1276D+02   1.7844D+00  0.0          2.7435D+05   0.0          5.5323D+04   1.5142D+03   1.7867D+01   0.0          1.5967D+03   1.0000D+00
             BOUNDARY VALUES
                                      0.0          0.0          5.5000D+04   1.5488D+03   0.0          0.0
```